

การหาค่าเหมาะสมที่สุดแบบจุดหมุนและการประยุกต์

THE PIVOT OPTIMIZER AND APPLICATION



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2564

KMITL-2021-EN-D-018-030

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE PIVOT OPTIMIZER AND APPLICATION



SOMSIN THONGKRAIRAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTORAL OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2021
KMITL-2021-EN- D-018-030

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2021

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การหาค่าเหมาะสมที่สุดแบบจุดหมุนและการประยุกต์
นักศึกษา	นายสมสิน ทองไกรรัตน์
รหัสประจำตัว	60601167
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2564
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.วันวิสา ชัชวงษ์

บทคัดย่อ

วิทยานิพนธ์นี้จัดทำขึ้นเพื่อนำเสนออัลกอริทึมการหาค่าเหมาะสมที่สุดที่ผู้วิจัยได้วิจัยและพัฒนาขึ้นมา มีชื่อว่า การหาค่าที่เหมาะสมแบบจุดหมุน หรือ pivot optimizer ซึ่งเป็นอัลกอริทึมที่ได้รับแรงบันดาลใจมาจากการดำเนินการทางเรขาคณิต ประสิทธิภาพ ข้อดี และข้อด้อยของอัลกอริทึม ถูกทดสอบด้วยฟังก์ชันมาตรฐานพร้อมด้วยเปรียบเทียบผลลัพธ์ดังกล่าวกับอัลกอริทึมอื่นที่นิยมใช้ในปัจจุบัน ผลลัพธ์ที่ออกมา นั้นถือเป็นที่น่าพอใจเนื่องจากให้คำตอบที่มีค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานที่ดีเมื่อเปรียบเทียบกับอัลกอริทึมอื่น ทั้งปัญหาที่มีความซับซ้อนสูงและความซับซ้อนต่ำ อีกทั้งความซับซ้อนของอัลกอริทึมยังอยู่ในเกณฑ์ที่ดีทำให้สามารถประมวลผลได้อย่างรวดเร็ว

เนื้อหาในวิทยานิพนธ์เล่มนี้ได้รวบรวมทฤษฎีบทที่เกี่ยวข้องกับอัลกอริทึมดังกล่าวเพื่อ อธิบายแนวคิดและ กลไกการทำงานของงานวิจัยชิ้นนี้ รวมถึงรวบรวมและวิเคราะห์วรรณกรรมในปัจจุบันที่เกี่ยวข้อง เพื่อให้ผู้อ่านได้รู้ถึง แนวทางการวิจัยและอัลกอริทึมรวมไปถึงการประยุกต์ต่างๆ ที่ มีการนำหลักการการหาค่าที่เหมาะสมที่สุดไปใช้งาน และรวบรวมงานวิจัยที่ผ่านมาของผู้วิจัยซึ่งได้ทำการทดลองเพิ่มประสิทธิภาพในด้านต่างๆ ของอัลกอริทึม

จากนั้นผู้วิจัยได้นำเสนอการนำอัลกอริทึมตระกูลดังกล่าวไปประยุกต์เพื่อปรับปรุงเส้นทางการเคลื่อนที่ของเครื่องจักร เพื่อให้การเคลื่อนที่เป็นไปอย่างมีประสิทธิภาพที่สุดโดยได้นำอัลกอริทึมของผู้วิจัยและอัลกอริทึมอื่น ๆ เป็นเครื่องมือในการปรับปรุงประสิทธิภาพ ซึ่งทำให้การเคลื่อนที่ของเครื่องจักรนั้นออกมาเป็นที่น่าสนใจตามประสิทธิภาพของอัลกอริทึมที่ผู้วิจัยได้คิดค้นขึ้นมา

Title	The Pivot optimizer and application
Student	Mr. Somsin Thongkrait
Student ID.	60601167
Degree	Doctoral of Engineering
Program	Electrical Engineering
Year	2021
Thesis Advisor	Asst. Prof. Dr. Vanvisa Chutchavong

ABSTRACT

This thesis proposes the new optimization algorithm that inspired by geometric method called “Pivot Optimizer”. Performance is determined by benchmarking with standard test function. The properties advantages and disadvantages are described by the result analyses and compare with another algorithm. the result is very satisfied by reason of better output and standard diversion and the complexity of computation are lower enough to pass the competition standard.

This thesis also presents the application of optimization algorithm to improve machine moving path. The serval number of target points are given as the optimization problem and use above algorithm to optimize machine path. The total time of running time is compared with another. Pivot optimization also perform outstanding performance in the section.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้และงานวิจัยสำเร็จลุล่วงไปได้ด้วยดี โดยการชี้นำที่ยอดเยี่ยมของ ผศ.ดร. วันวิสา ชัชวรงค์ ผู้เป็นอาจารย์ที่ปรึกษาซึ่งคอยแนะนำ วิธีการดำเนินงานวิจัย ให้ความรู้ ตลอดจนการ ตีพิมพ์ผลงานวิจัย และเป็นหัวหน้างานที่ใจดีเสมอมา

ขอขอบคุณพี่เห่ง แห่งห้องแลป RailLink และพี่พี แห่งห้องแล็บชั้น 11 ที่ช่วยแนะนำ หนังสือดี ๆ และงานประชุมทางวิชาการเรียกได้ว่าเป็นที่ปรึกษาทางวิชาการให้แก่ผม และเพื่อน ๆ ทุกคนในห้องแลป

ขอขอบคุณปอน และ อีเมลล์ ที่ช่วยงานเกี่ยวกับ “การหาค่าที่เหมาะสมที่สุด” และร่วม ศึกษางานด้านนี้จนออกมาเป็นงานวิจัยที่สามารถเผยแพร่ได้ในที่สุด

ขอขอบคุณ อ.บุญยชนะ ภูระหงษ์ ที่แนะนำการศึกษาต่อ และเป็นผู้แนะนำให้รู้จักเทคโนโลยี ใหม่ ๆ เพื่อเป็นแนวทางในการวิจัย

สุดท้ายนี้ขอขอบคุณครอบครัวของผมทั้ง คุณพ่อ คุณแม่ พี่น้อง และญาติคนอื่น ๆ ที่มีส่วนให้ ผมมีโอกาสได้ศึกษาต่อในสถาบันแห่งนี้ และสนับสนุนค่าเล่าเรียน รวมทั้งเป็นกำลังในการทำงาน วิจัยในขณะที่ศึกษา

สมสิน ทองไกรรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ III อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
รายการคำย่อและสัญลักษณ์.....	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	2
1.3 สมมุติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัยวิธีการวิจัย.....	2
1.5 ขอบเขตการวิจัย.....	3
1.6 ขั้นตอนของการศึกษา.....	3
1.7 เครื่องมือและอุปกรณ์ที่ใช้ในการวิจัย.....	3
1.8 ส่วนประกอบของวิทยานิพนธ์.....	4
บทที่ 2 ความรู้พื้นฐานที่เกี่ยวข้องกับงานวิจัย.....	5
2.1 อัลกอริทึมการหาค่าเหมาะสมที่สุด (Optimization Algorithm)	5
2.2 อัลกอริทึมการวิวัฒนาการ (Evolutionary Algorithm)	7
2.3 การคิดระยะทางแบบ Mahalanobis (Mahalanobis Distance)	8
2.4 การแจกแจงแบบเอกรูปและแบบปกติ (Uniform and Normal Distribution).....	9
บทที่ 3 งานวิจัยที่เกี่ยวข้อง	12
3.1 วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization, PSO)	12
3.2 วิธีหาค่าเหมาะสมที่สุดแบบฝูงหมาป่าเทา (Grey Wolf Optimizer, GWO).....	14
3.3 วิธีหาค่าเหมาะสมที่สุดแบบวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution, DE)	17
3.4 วิธีหาค่าเหมาะสมที่สุดแบบรังผึ้งสังเคราะห์ (artificial bee colony, ABC).....	19
3.5 การวิเคราะห์และการเปรียบเทียบประสิทธิภาพของอัลกอริทึมในรูปแบบของ CEC.....	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ IV อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.6 การผสมผสานกลศาสตร์การเคลื่อนไหวเพื่อปรับปรุงเส้นทางของเครื่องจักรแบบหัวฉัด (Incorporating Kinematic Properties into Fused Deposition Toolpath Optimization).....	22
3.7 การหาเวลาที่ดีที่สุดของการกัดชิ้นงานโดยมีความเร่งเป็นเงื่อนไข (Minimum Time Feed-rate Optimization along Predefined Toolpath with Acceleration Constraints on Each Axis)	23
3.8 การใช้อัลกอริทึม PSO เพื่อกำหนดอัตราการกินวัสดุที่เหมาะสมที่สุดบนพื้นฐานของการเขียนแบบ NURBS (PSO based feedrate optimization with contour error constraints for NURBS toolpaths).....	24
บทที่ 4 การเพิ่มประสิทธิภาพให้กับอัลกอริทึมการหาค่าที่เหมาะสมที่สุด	25
4.1 บทนำ.....	25
4.2 การเพิ่มประสิทธิภาพด้านเวลาของการหาค่าเหมาะสมที่สุดแบบแบ่งกลุ่มอนุภาคโดยใช้การประมวลผลแบบขนาน (A time improvement PSO base algorithm using multithread programming).....	25
4.2.1 ปัญหาการเดินทางของนักขาย (Traveling salesman problem: TSP)	26
4.2.2 การแก้ไขปัญหาด้วยวิธี 2-opt swap	27
4.2.3 การประยุกต์ใช้อัลกอริทึม 2-opt ด้วย PSO.....	27
4.2.4 การคำนวณ PSO แบบ Multithread.....	28
4.2.5 การทดลอง.....	29
4.3 การประมวลผลค่าที่เหมาะสมที่สุดแบบไม่สุ่มเพื่อคำตอบที่คงที่ (Hard Deterministic Particle Swarm Optimization for Certain Result Solution, HD-PSO)	31
4.3.1 การหาค่าเหมาะสมแบบอนุภาคที่สุ่มโดยไม่สุ่ม	32
4.3.1.1 การแบ่งขอบ (Bound slicing).....	32
4.3.1.2 การใช้หน้าต่างเคลื่อนที่ (Windows moving).....	34
4.3.1.3 การผ่าเหล่าแบบตามลำดับ (Sorting mutation).....	35
4.3.2 การทดลองและผลลัพธ์	36
4.3.2.1 ลักษณะเฉพาะของ HD-PSO	36
4.3.2.2 ประสิทธิภาพของอัลกอริทึม HD-PSO.....	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ V ังอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 การหาค่าที่เหมาะสมที่สุดแบบจุดหมุน (Pivot Optimiser).....	40
5.1 บทนำ.....	40
5.2 จุดหมุน (Pivot).....	41
5.3 การสร้างประชากรใหม่ (Offspring Creation)	42
5.4 การคัดสรรประชากร (Selection)	44
5.5 การชดเชยประชากร (Population Compensation)	46
5.6 การทำงานของอัลกอริทึม (Process Flow)	47
5.7 การทดลองประสิทธิภาพของ Pivot optimizer.....	48
5.8 ผลการทดลอง.....	50
5.9 การวิเคราะห์ผลการทดลอง.....	57
5.10 สรุปผลการทดลอง.....	58
บทที่ 6 การประยุกต์ใช้งานคำนวณเส้นทางการเคลื่อนที่.....	59
6.1 บทนำ.....	59
6.2 ปัญหา PTP motion optimization	61
6.3 การทดลองใช้การหาค่าที่เหมาะสมที่สุดเพื่อแก้ไขปัญหา	70
6.4 ผลการทดลอง.....	71
6.5 สรุปผลการทดลอง.....	73
บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ.....	75
7.1 สรุปผลการวิจัย.....	75
7.2 ข้อเสนอแนะ.....	75
เอกสารอ้างอิง.....	77
ภาคผนวก.....	80
ภาคผนวก ก. ฟังก์ชันทดสอบที่ใช้ในการทดลอง (Test Function).....	81
ภาคผนวก ข. โจทย์ปัญหาการเคลื่อนที่ในการทดลอง (Motion Problem Set).....	94
ภาคผนวก ค. ผลงานวิจัยที่ได้รับการตีพิมพ์.....	97
ประวัติผู้เขียน.....	98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ VI อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 ผลลัพธ์ของฟังก์ชันทดสอบต่าง ๆ ใน 3 มิติ	38
4.2 ผลลัพธ์ของฟังก์ชันทดสอบต่าง ๆ ใน 4 มิติ	39
5.1 ฟังก์ชันทดสอบที่ใช้ในการทดลองและคุณสมบัติ	48
5.2 ผลการทดลองกับฟังก์ชันทดสอบที่ 10 มิติ	50
5.3 ผลการทดลองกับฟังก์ชันทดสอบที่ 50 มิติ	52
5.4 ผลการทดลองกับฟังก์ชันทดสอบที่ 100 มิติ	55
6.1 ผลการทดลองปัญหาการเคลื่อนที่	72



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ VII อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ค่าสูงสุดและค่าต่ำสุดของฟังก์ชัน.....	5
2.2 ฟังก์ชันที่เปลี่ยนไปโดยการบวกหรือคูณแต่ค่าต่ำสุดยังคงเดิม.....	6
2.3 Pseudo code ของอัลกอริทึมวิวัฒนาการทั่วไป	7
2.4 เปรียบเทียบระยะห่าง MD และระยะห่างแบบยูคลิด.....	9
2.5 เปรียบเทียบความน่าจะเป็นระหว่างการกระจายแบบเอกรูปและการกระจายแบบปกติ.....	11
3.1 ตำแหน่งของอนุภาคในอัลกอริทึม PSO ในรุ่นที่แตกต่างกัน.....	13
3.2 ตำแหน่งที่เป็นไปได้ในระยะการล้อมเหยื่อของอัลกอริทึม GWO	15
3.3 ตำแหน่งที่เป็นไปได้ในระยะการล่าเหยื่อของอัลกอริทึม GWO	16
3.4 ตัวแปร A ที่กำหนดการโจมตีเหยื่อและการหาเหยื่อตัวถัดไป	16
3.5 ขั้นตอนการสร้างประชากรตัวใหม่ของอัลกอริทึมวิวัฒนาการโดยใช้ผลต่าง	17
3.6 ฟังก์ชันการทำงานของอัลกอริทึมวิวัฒนาการโดยใช้ผลต่าง.....	18
3.7 pseudo code ของอัลกอริทึม artificial bee colony	20
3.8 ปัญหาแบบพื้นฐานนิยมเดียวและพหุนิยมใน 1 มิติ.....	21
3.9 กราฟเทียบแสดงสถิติระหว่างอัลกอริทึม LS และการ slicing ทั่วไป.....	22
3.10 กราฟเปรียบเทียบความเร่งระหว่างการคิดเส้นทางแบบไม่กำหนดความเร่งและแบบกำหนดความเร่ง	23
3.11 ขั้นตอนการปรับปรุงเส้นทางของเครื่องจักรและความเร็วที่ใช้.....	24
4.1 ตัวอย่างปัญหาการเดินทางของนักขาย.....	26
4.2 รหัสเทียมของการประยุกต์ใช้อัลกอริทึม 2-opt ด้วย PSO	27
4.3 การสุ่มสร้างคำตอบใหม่ของอัลกอริทึม 2-opt ด้วย PSO	28
4.4 ตัวอย่างปัญหาการเดินทางของนักขาย.....	29
4.5 ตัวอย่างของชุดข้อมูลปัญหา (ja9847).....	30
4.6 แสดงคำตอบของระบบของชุดคำถาม (zi929) ของอัลกอริทึม PSO และ 2-opt	30
4.7 กราฟแสดงเวลาและผลลัพธ์ที่ได้จากการคำนวณ 2-opt.....	31
4.8 รหัสเทียมของการแบ่งของ.....	33
4.9 ตัวอย่างการแบ่งขอบในปัญหา 3 มิติ	34
4.10 แสดงการทำงานของการทำงานใช้หน้าตาที่เคลื่อนที่.....	34
4.11 การเลือกประชากรในรุ่นต่อไป.....	35

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.12 ผังงานแสดงการทำงานของทั้งระบบ.....	36
4.13 ตำแหน่งของอนุภาคเมื่อเริ่มทำงาน.....	37
4.14 กราฟเปรียบเทียบค่าคำตอบระหว่าง HD-PSO และ PSO แบบธรรมดา.....	37
5.1 จำนวนและประเภทของประชากรใหม่.....	42
5.2 กราฟเปรียบเทียบค่าคำตอบระหว่าง HD-PSO และ PSO แบบธรรมดา.....	43
5.3 แนวโน้มของการหาคำตอบใหม่	44
5.4 การหาจุดหมุนขอบและจุดหมุนไม่ซ้ำซ้อน.....	45
5.5 การลบและเติมประชากรหลังจากการคัดสรร.....	46
5.6 การสร้างกลุ่มประชากรที่มีความเป็นไปได้	47
5.7 รหัสเทียมของการหาค่าเหมาะสมที่สุดแบบจุดหมุน	48
6.1 จุดกลางเครื่องมือ (TCP) ของเครื่องจักร	59
6.2 รูปแบบการเคลื่อนที่แบบ PTP(MoveJ) และ LIN(MoveL).....	60
6.3 ตัวอย่างโปรแกรมใน Robot programming language (KUKA version.).....	60
6.4 การกำหนดพิกัดของเครื่องจักรที่มี 6 แกนอิสระ	62
6.5 เส้นทางที่เกิดเมื่อตั้งความเร็วเมื่อผ่านจุดให้มีความสูงและเส้นทางที่เกิดเมื่อตั้งความเร็วเมื่อผ่านจุดให้มีความต่ำ.....	63
6.6 เส้นทางเคลื่อนที่และการตั้งค่าที่ได้จากการหาค่าที่เหมาะสมที่สุด.....	63
6.7 ความเร็วตลอดการเคลื่อนที่จากจุดเริ่มต้นไปยังจุดเป้าหมาย.....	64
6.8 กรณีของการเคลื่อนที่ต่าง ๆ ที่สามารถเกิดขึ้นได้.....	65
6.9 ตัวอย่างการเลือกเวลาในการเคลื่อนที่ ระหว่างเป้าหมาย	66
6.10 การเคลื่อนที่ที่ไม่มีกรณีสาม	67
6.11 การเคลื่อนที่ที่มีเพียงกรณีแรก	67
6.12 ตัวอย่างคำถามและคำตอบตั้งต้นของเส้นทางเคลื่อนที่	69
6.13 การพิจารณาคำตอบตั้งต้น.....	69
6.14 ตัวอย่างการหาคำตอบของปัญหาการเคลื่อนที่	72
6.15 กราฟเปรียบเทียบคำตอบของแต่ละวิธีการ	73
ก1.1 ฟังก์ชัน Rosenblock ในระนาบ 2 มิติ	84
ก1.2 ฟังก์ชัน Rastrigin ในระนาบ 2 มิติ.....	85
ก1.3 ฟังก์ชัน Schwefel ในระนาบ 2 มิติ.....	85

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก1.4 ฟังก์ชัน Ackley ในระนาบ 2 มิติ.....	86
ก1.5 ฟังก์ชัน Griewank ในระนาบ 2 มิติ.....	86
ก1.6 ฟังก์ชัน Rotated Bent Cigar ในระนาบ 2 มิติ.....	87
ก1.7 ฟังก์ชัน Rotated Discus Function ในระนาบ 2 มิติ.....	87
ก1.8 ฟังก์ชัน Shifted and Rotated Rosenbrock ในระนาบ 2 มิติ.....	88
ก1.9 ฟังก์ชัน Shifted and Rotated Ackley ในระนาบ 2 มิติ.....	88
ก1.10 ฟังก์ชัน Shifted and Rotated Weierstrass ในระนาบ 2 มิติ.....	89
ก1.11 ฟังก์ชัน Shifted and Rotated Ackley ในระนาบ 2 มิติ.....	89
ก1.12 ฟังก์ชัน Shifted and Rotated Katsuura ในระนาบ 2 มิติ.....	90
ก1.13 ฟังก์ชัน Shifted and Rotated Weierstrass ในระนาบ 2 มิติ.....	90
ก1.14 ฟังก์ชัน Shifted and Rotated HGBat ในระนาบ 2 มิติ.....	91
ก1.15 ฟังก์ชัน Shifted and Rotated Expanded Griewank plus Rosenbrock ในระนาบ 2 มิติ.....	91
ก1.16 ฟังก์ชัน Shifted and Rotated Expanded Scaffer ในระนาบ 2 มิติ.....	92
ข1.1 ปัญหาการเคลื่อนที่ลำดับที่ 1 ถึง 5.....	94
ข1.2 ปัญหาการเคลื่อนที่ลำดับที่ 6 ถึง 8.....	95
ข1.3 ปัญหาการเคลื่อนที่ลำดับที่ 9 และ 10.....	96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ X อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการคำย่อและสัญลักษณ์

การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization)	PSO
การหาค่าเหมาะสมที่สุดแบบฝูงหมาป่าเทา (Grey Wolf Optimizer)	GWO
การหาค่าเหมาะสมที่สุดแบบวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution)	DE
การหาค่าเหมาะสมที่สุดแบบรังผึ้งสังเคราะห์ (artificial bee colony)	ABC
Evolutionary Algorithm	EA
Mahalanobis Distance	MD
IEEE Congress on Evolutionary Computation	CEC
ปัญหาการเดินทางของนักขาย (Traveling salesman problem)	TSP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ **XI** อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การค้นหาค่าที่เหมาะสมที่สุดหรือ Optimization มีส่วนสำคัญในการวิจัย พัฒนา และการทดลองทฤษฎีต่างๆ ในปัจจุบันไม่ว่าจะเป็นในด้านของการแก้ไขปัญหาลักษณะยาก (NP-Problem) การสร้างแบบจำลอง (Simulation) หรือแม้แต่การพยากรณ์ความเสี่ยงทางการเงิน (Economics and finance) เนื่องจากปัญหาต่างๆ ที่ได้ยกตัวอย่างมานั้นไม่สามารถแก้ไขได้โดยวิธีโดยตรง (Direct method) ในยุคปัจจุบัน อันเนื่องมาจาก ความหลากหลายของคำตอบที่เป็นไปได้ทั้งหมดมีจำนวนมากเกินขีดความสามารถของอุปกรณ์คำนวณในปัจจุบัน เช่น คอมพิวเตอร์ การที่จะได้มาซึ่งคำตอบที่ดีที่สุด (Best solution) ของแต่ละปัญหานั้น ยังเป็นความท้าทายทางการวิจัย และยังคงต้องพัฒนาต่อไป แต่หากเป้าหมายของการใช้งานนั้นต้องการคำตอบที่เหมาะสมเพื่อที่จะนำไปใช้งาน และสามารถละลายกฎเกณฑ์ที่ว่าคำตอบที่ได้มานั้นจะต้องเป็นคำตอบที่ดีที่สุดของปัญหาวิธีการค้นหาค่าที่เหมาะสมที่สุด สามารถนำมาใช้เพื่อหาคำตอบที่ดีและเหมาะสมกับการใช้งาน (Optimal solution) แทนคำตอบที่ดีที่สุดได้

คำตอบที่เหมาะสมที่มาจากวิธีการค้นหาค่าที่เหมาะสมที่สุดนั้นจะขึ้นอยู่กับวิธีหรือเครื่องมือที่ใช้ในการค้นหา (Optimization algorithm) หรือเรียกสั้น ๆ ว่า Optimizer กระบวนการดังกล่าวได้มีการวิจัยกันมาอย่างยาวนาน และแต่ละกระบวนการนั้นย่อมมีจุดกำเนิดที่แตกต่างกันไป เช่น พฤติกรรมของสัตว์ (Social behavior inspired) ทฤษฎีในการคำนวณ (Mathematical base inspired) เป็นต้น ซึ่งมีประสิทธิภาพแตกต่างกันไปตามการออกแบบ

เครื่องมือการหาค่าที่เหมาะสมที่ดั่งนั้นควรจะให้ผลลัพธ์ที่ใกล้เคียงกับผลลัพธ์ที่ดีที่สุด อีกทั้งควรใช้เวลาในการหาผลลัพธ์ให้เร็วและใช้ทรัพยากรให้น้อยอีกด้วย จึงเป็นวิธีการที่สามารถนำไปประยุกต์ใช้ได้จริงได้ดีที่สุด ผู้วิจัยจึงได้คิดค้นวิธีการหาค่าที่เหมาะสมที่สุดขึ้นมาโดยใช้แรงบันดาลใจจากทฤษฎีทางเรขาคณิตหลายมิติ (Multi-dimensional Geographic) มาประยุกต์ทำให้เกิดเครื่องมือการหาค่าที่เหมาะสมที่สุดวิธีใหม่ในชื่อว่า การหาค่าที่ดีที่สุดแบบจุดหมุน (Pivot optimizer) ซึ่งมีความสามารถในด้านการหาคำตอบที่ไม่ซ้ำกัน และออกแบบมาให้ใช้ทรัพยากรประหยัดที่สุด เพื่อความเป็นไปได้ในการนำไปประยุกต์ใช้จริง (Feasibility)

เพื่อที่จะวัดประสิทธิภาพของอัลกอริทึมที่คิดค้นขึ้นมาใหม่นี้ ผู้วิจัยได้นำฟังก์ชันต่าง ๆ ที่เป็นที่รู้จักกันว่าเป็นฟังก์ชันในการทดสอบ (Well-known Test function) มาทดสอบกับแนวคิดใหม่นี้และทำการเปรียบเทียบกับอัลกอริทึมอื่นที่เป็นที่นิยมใช้ในปัจจุบันและที่ผ่าน ๆ มา เพื่อเปรียบเทียบและยืนยันประสิทธิภาพของวิธีการใหม่นี้ อีกทั้งผู้วิจัยได้ทำการประยุกต์ทฤษฎีดังกล่าว กับโจทย์ปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุงการเคลื่อนที่แบบจุดต่อจุด ของเครื่องจักร (Point-to-point motion) เพื่อเพิ่มประสิทธิภาพในด้านของเวลา เป็นการพิสูจน์ความสามารถในหลาย ๆ ด้านของอัลกอริทึมใหม่นี้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้จัดทำขึ้นเพื่อนำเสนออัลกอริทึมการหาค่าที่เหมาะสมที่สุดแบบจุดหมุน หรือ Pivot optimization algorithm ซึ่งเป็นอัลกอริทึมหรือกระบวนการการคิดใหม่ที่ผู้วิจัยเป็นผู้คิดค้นขึ้นมา โดยจะทำการนำเสนอหลักการทำงานทั้งหมดของอัลกอริทึมนี้ วิธีการประยุกต์ใช้ได้ พร้อมด้วยผลการทดลองและบทวิเคราะห์การทดลองที่แสดงให้เห็นถึงประสิทธิภาพในการทำงานเมื่อเทียบกับอัลกอริทึมอื่น ๆ ที่มีอยู่ในปัจจุบัน และสาธิตการนำไปประยุกต์ ในไมโครคอนโทรลเลอร์ (Microcontroller) ที่มีทรัพยากรอยู่อย่างจำกัด

1.3 สมมุติฐานของการศึกษา

อัลกอริทึมที่ได้คิดค้นขึ้นมาใหม่นี้หรือ Pivot optimizer จะใช้ประโยชน์จากทฤษฎีทางเรขาคณิตหลายมิติ (Multi-dimensional Geographic) เพื่อให้ได้มาซึ่งคำตอบที่เหมาะสมที่สุด โดยจะมีค่าเทียบเท่าหรือใกล้เคียงกับคำตอบที่ดีที่สุดของปัญหาที่อยู่ในการทดลอง และสามารถประยุกต์ลงในทุกระบบได้อย่างมีประสิทธิภาพ แม้กระทั่งระบบที่มีทรัพยากรจำกัด

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

การออกแบบอัลกอริทึมของผู้วิจัยรวมไปถึงการออกแบบการทดลองนั้นได้แนวคิดมาจากการแข่งขันด้านการหาค่าที่เหมาะสมที่สุด (CEC Competitions) ซึ่งเป็นหัวข้อหนึ่งในการประชุมทางวิชาการทางด้านการหาค่าที่เหมาะสมอยู่เสมอ เช่น IEEE WCCI หรือ IEEE Congress on Evolutionary Computation ซึ่งจะจัดขึ้นในทุกปี

ฟังก์ชันในการทดลองส่วนใหญ่ที่ผู้วิจัยเลือกมาจะเป็นฟังก์ชันเดียวกันกับที่ใช้ในการแข่งขัน และเพื่อความเท่าเทียมในการเปรียบเทียบผลการทดลอง (Fairness) ผู้วิจัยได้เลือกแพลตฟอร์ม (Platform) หรือ สภาพแวดล้อมในการทดลองที่เป็นสากลมีชื่อว่า pagmo เพื่อไม่ให้เป็นการเข้าข้างผลการทดลองของผู้วิจัยจนเกินไป

1.5 ขอบเขตการวิจัย

ขอบเขตของการวิจัยนี้สามารถอธิบายได้ดังต่อไปนี้

1.5.1 ปัญหาทั้งหมดที่อยู่ในขอบเขตการวิจัยจะเป็นปัญหาประเภทการหาค่าที่เหมาะสมที่สุดของตัวแปรเดียวแบบมีข้อจำกัด (Unconstrained single variable optimization) เท่านั้น

1.5.2 การทดสอบอัลกอริทึมด้วยฟังก์ชันต่างๆ จะมีข้อกำหนดที่เท่าเทียมกันเช่น จำนวนรอบในการค้นหาคำตอบหรือ Function Evaluation Count และจะบันทึกเป็นสถิติ เช่น กรณีที่ดีที่สุด กรณีที่แย่ที่สุด เพื่อใช้ในการวิเคราะห์ผล

1.5.3 ฟังก์ชันทดสอบต่างๆ จะเป็นฟังก์ชันทั่วไป ไม่ได้เป็นฟังก์ชันที่ออกแบบมาเพื่ออัลกอริทึมใดอัลกอริทึมหนึ่ง

1.5.4 ผลการทดลองที่ออกมาไม่รับรองว่าอัลกอริทึมของผู้วิจัยจะดีกว่าอัลกอริทึมอื่นเสมอ เนื่องจากอัลกอริทึมแต่ละอัลกอริทึม มีข้อดีข้อเสียที่แตกต่างกันไป

1.6 ขั้นตอนของการศึกษา

ขั้นตอนการศึกษางานวิจัยนี้สามารถอธิบายได้ดังต่อไปนี้

1.6.1 ศึกษาความรู้เกี่ยวกับการหาค่าที่เหมาะสมที่สุด

1.6.2 ทบทวนวรรณกรรมอัลกอริทึมการหาค่าที่เหมาะสมที่เป็นที่นิยมในปัจจุบันผ่านทางงานวิจัย

1.6.3 วิเคราะห์ข้อดีข้อเสียของอัลกอริทึมและงานวิจัยที่เกี่ยวข้อง สรุปผลและพิจารณาแนวทางการทำวิจัย

1.6.4 จำลองการทำงานของอัลกอริทึมแต่ละตัวในสภาพแวดล้อมของระบบ บนคอมพิวเตอร์ของผู้วิจัย

1.6.5 ค้นคว้าหาทฤษฎีที่สามารถประยุกต์ในการหาค่าที่เหมาะสมที่สุด

1.6.6 ออกแบบอัลกอริทึม Pivot optimizer

1.6.7 ประยุกต์แนวคิดของอัลกอริทึมที่ออกแบบมาพร้อมเปรียบเทียบประสิทธิภาพกับอัลกอริทึมอื่น

1.6.8 สรุปผลและปรับปรุงข้อด้อยของอัลกอริทึม

1.6.9 ตีพิมพ์ผลงานในวารสารการประชุมทางวิชาการ

1.7 เครื่องมือและอุปกรณ์ที่ใช้ในการวิจัย

1.7.1 คอมพิวเตอร์ระบบปฏิบัติการ Microsoft windows

1.7.2 ซอฟต์แวร์ pagmo (rev 2.16.1) เพื่อการจำลองผล

1.7.3 ซอฟต์แวร์ Microsoft visual studio (C#) (rev. 16.8.3) เพื่อการแสดงผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.8 ส่วนประกอบของวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้ ประกอบด้วย 7 บท ได้แก่

บทที่ 1 บทนำ

บทที่ 2 ความรู้พื้นฐานที่เกี่ยวข้องกับงานวิจัย

บทที่ 3 งานวิจัยที่เกี่ยวข้อง

บทที่ 4 การเพิ่มประสิทธิภาพให้กับอัลกอริทึมการหาค่าที่เหมาะสมที่สุด

บทที่ 5 การหาค่าที่เหมาะสมที่สุดแบบจุดหมุน (Pivot Optimiser)

บทที่ 6 การประยุกต์ใช้งานคำนวณเส้นทางการเคลื่อนที่

บทที่ 7 สรุปผลการวิจัย และข้อเสนอแนะ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

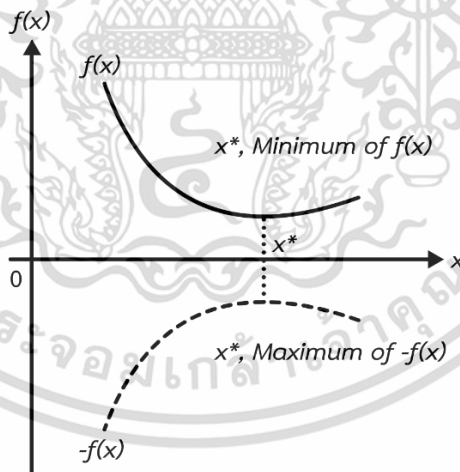
บทที่ 2

ความรู้พื้นฐานที่เกี่ยวข้องกับงานวิจัย

2.1 อัลกอริทึมการหาค่าเหมาะสมที่สุด (Optimization Algorithm)

การหาค่าเหมาะสมที่สุดคือกระบวนการที่ได้มาซึ่งคำตอบที่ 'ดีที่สุด' ภายใต้สถานการณ์ใด สถานการณ์หนึ่ง การออกแบบและการคำนวณทางวิศวกรรมส่วนใหญ่มีกระบวนการที่ต้องเลือกและตัดสินใจค่าต่าง ๆ ที่ใช้ในการคำนวณ เป้าหมายของการ 'เลือก' ที่ดีนั้นคือการทำให้ผลลัพธ์ออกมามีประสิทธิภาพมากที่สุด กล่าวคือ การได้มาซึ่งผลลัพธ์ที่มีประโยชน์ 'มากที่สุด' ในขณะที่ออกแรงทำงานไป 'น้อยที่สุด'

หากจะกล่าวถึงในเชิงการหาค่าที่เหมาะสมที่สุดเราจะจัดให้ แรงที่เราต้องออกเพื่อทำงานและผลลัพธ์นั้นอยู่ในรูปของฟังก์ชันเป้าหมาย (Objective function) ของตัวแปรการตัดสินใจ (Decision variable) การหาค่าที่เหมาะสมที่สุด นั้น จะหมายถึงการหาเงื่อนไขการทำงานที่จะทำให้ฟังก์ชันเป้าหมายมีค่าที่มากที่สุดหรือน้อยที่สุด รูปที่ 2.1 แสดงให้เห็นถึงจุด x^* ซึ่งเป็นจุดที่ทำให้ฟังก์ชัน $f(x)$ มีค่าน้อยที่สุด และในขณะเดียวกันยังทำให้ $-f(x)$ มีค่ามากที่สุดอีกด้วย ดังนั้นโดยทั่วไปแล้วการได้มาซึ่งค่าที่น้อยที่สุดจะทำให้รู้ถึงค่าที่มากที่สุดมาด้วยโดยการทำให้ฟังก์ชันนั้นมีค่าเป็นลบ



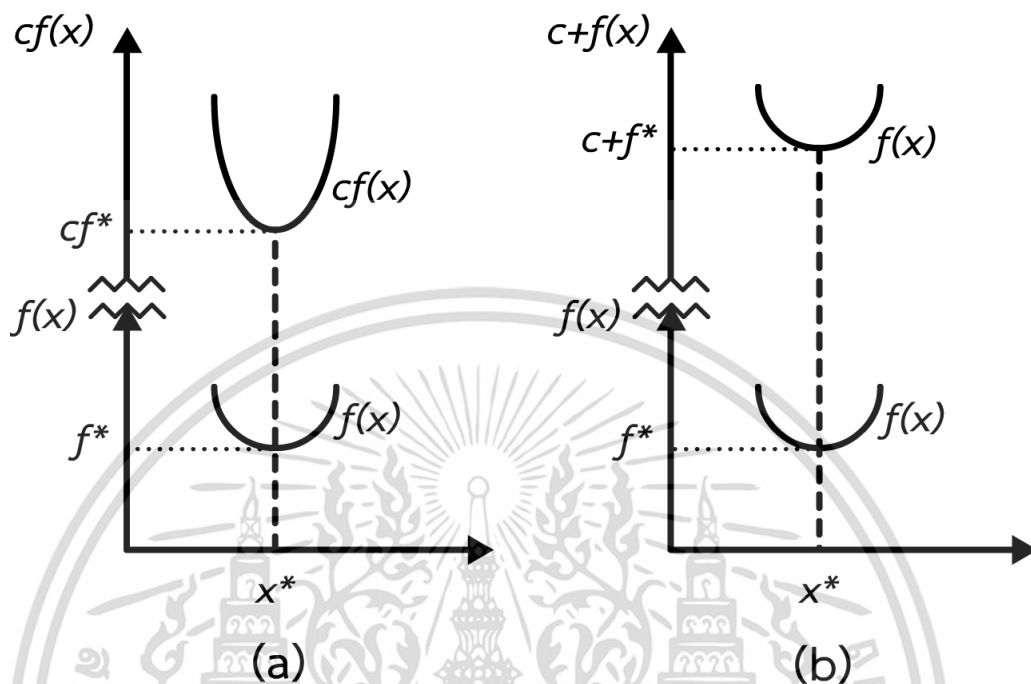
รูปที่ 2.1 ค่าสูงสุดและค่าต่ำสุดของฟังก์ชัน

นอกจากนั้น การกระทำต่อไปนี้ต่อฟังก์ชันเป้าหมายจะไม่ทำให้ค่าที่เหมาะสมที่สุดของฟังก์ชันเปลี่ยนแปลงไปได้

- การคูณหรือการหาร (Multiplication or Division) ของฟังก์ชันด้วยจำนวนเต็มบวกคงที่ c (แสดงในรูปที่ 2.2a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การบวกหรือการลบ (Addition or Subtraction) ของฟังก์ชันด้วยจำนวนเต็มบวกคงที่ c (แสดงในรูปที่ 2.2b)



รูปที่ 2.2 ฟังก์ชันที่เปลี่ยนไปโดยการบวกหรือคูณแต่ค่าต่ำสุดยังคงเดิม

กระบวนการหรืออัลกอริทึมการหาค่าที่เหมาะสมที่สุดนั้นได้ถูกวิจัยและพัฒนาขึ้นมาอย่างแพร่หลาย เนื่องจากวิธีการที่คิดค้นขึ้นมาแต่ละวิธีนั้นมีจุดประสงค์การพัฒนาที่แตกต่างกันออกไปตามประเภทของปัญหาที่ต้องการจะแก้ไข และไม่มีวิธีการใดที่สามารถแก้ไขปัญหาคอบคลุมได้ทุกปัญหา [1]

หากจะแบ่งประเภทของอัลกอริทึมการหาค่าที่เหมาะสมที่สุดตามการทำงานแล้วจะสามารถแบ่งได้ 3 ตระกูลหลัก ได้แก่

- Analytical search method คือ การใช้วิธีทางคณิตศาสตร์ (Mathematical programming) ในการหาค่าที่ต่ำที่สุด วิธีนี้จะใช้คุณสมบัติต่าง ๆ ทางคณิตศาสตร์ของฟังก์ชันเป็นตัวนำทางในการค้นหา เช่น ความชันของฟังก์ชัน (Gradient)
- Enumeration search method คือการใช้การสุ่มโดยสุ่มทุก ๆ ความเป็นไปได้ของตัวแปรในฟังก์ชัน เพื่อหาค่าที่ต่ำที่สุด
- Heuristic search method คือการใช้ผลลัพธ์ของการคำนวณก่อนหน้าเป็นตัวตัดสินใจและคำนวณตัวแปรในการคำนวณครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการที่ได้กล่าวในข้างต้นอาจมีการใช้งานที่ผสมผสานกันได้ตามแต่อัลกอริทึมที่คิดค้นขึ้นมา [2]

2.2 อัลกอริทึมการวิวัฒนาการ (Evolutionary Algorithm)

อัลกอริทึมการวิวัฒนาการ หรือ Evolutionary Algorithm (EA) หมายถึงอัลกอริทึมที่ทำการแก้ไขปัญหาโดยยึดหลักการการพัฒนาผลลัพธ์ (Solution) ขึ้นไปซ้ำ ๆ หลายรอบ (Iteration) โดยปกติ การหาผลลัพธ์ใน 1 รอบของอัลกอริทึมนี้ จะเรียกว่า 1 รุ่น (Generation) เช่นเดียวกันกับคำศัพท์ในเชิงของชีววิทยา [3]

อัลกอริทึมตระกูลนี้จะกำหนดให้มีตัวแทนของการหาคำตอบ (Search agent) ได้หลายตัว เรียกว่าประชากร และจะกล่าวถึงการพัฒนาประชากรทั้งหมดในระบบ (Population base) ในแต่ละรอบของการพัฒนาผลลัพธ์ โดยการพัฒนาดังกล่าวจะใช้แนวคิดมาจากทฤษฎีการวิวัฒนาการทั้งสิ้น ซึ่งโดยทั่วไปแล้วจะใช้แนวคิดดังต่อไปนี้

- Mutation คือ การนำประชากรตัวใดตัวหนึ่งมาทำการเปลี่ยนแปลงคุณสมบัติบางอย่างโดยไม่ขึ้นอยู่กับประชากรตัวอื่นทำให้เกิดการเปลี่ยนแปลงที่คาดหวังว่าจะดีขึ้น
- Selection คือ แนวคิดในการเลือกประชากรที่ดีที่สุดมาเป็นต้นแบบในการสร้างประชากรรุ่นต่อไปที่มีผลลัพธ์ดีขึ้น
- Crossover คือ การนำประชากรตั้งแต่ 2 ตัวขึ้นไปมาแล้วเลือกเฉพาะคุณสมบัติที่ดีที่สุดของแต่ละตัวมาเป็นต้นแบบในการสร้างประชากรตัวใหม่
- Recombination คือ การสุ่มเลือกคุณสมบัติต่าง ๆ ของประชากรตัวหนึ่ง ๆ มาและทำการทดลองสลับตำแหน่งของคุณสมบัตินี้

นอกจากกระบวนการข้างต้นที่กล่าวถึงแล้ว ยังมีกระบวนการอื่น ๆ ที่ได้แนวคิดมาจากการวิวัฒนาการที่สามารถนำมาประยุกต์ในอัลกอริทึมตระกูลนี้ได้ การวิเคราะห์การทำงานของอัลกอริทึมวิวัฒนาการนั้นจะมีแบบแผนค่อนข้างตายตัวตาม Pseudo code ดังแสดงในรูปที่ 2.3

```

1 Generate the initial population randomly
2 Repeat following step until termination :
3   Evaluate the fitness of each individual
4   Select the fittest individual for reproduction
5   Breed new individual through crossover and mutation
6   Replace the least-fit individual with new individual

```

รูปที่ 2.3 Pseudo code ของอัลกอริทึมวิวัฒนาการทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากอัลกอริทึมประเภทนี้ใช้การพัฒนาผลลัพธ์ให้ดียิ่งขึ้นไปเรื่อย ๆ ปัจจัยภายนอกอื่น เช่น ความชันของฟังก์ชัน และ ค่าขอบของตัวแปร เป็นต้น จะไม่มีผลต่อการทำงาน ทำให้อัลกอริทึมประเภทนี้ใช้ได้ผลต่อปัญหาทุกประเภท

2.3 การคิดระยะทางแบบ Mahalanobis (Mahalanobis Distance)

ระยะทางแบบ Mahalanobis (Mahalanobis Distance : MD) คือระยะห่างระหว่างจุด 2 จุดในระบบพิกัดแบบหลายตัวแปร (Multivariate space) โดยปกติการคำนวณระยะห่างจะใช้ระยะห่างในระบบของปริภูมิยูคลิด (Euclidean space) ยกตัวอย่างเช่น ระบบพิกัดแบบ 3 ตัวแปรจะใช้พิกัดในแนวแกน X Y และ Z ในการอ้างอิงระยะห่างและสามารถหาระยะห่างออกมาได้เปรียบเสมือนการหาระยะห่างในชีวิตจริง ดังสมการที่ (2.1) แสดงการหาระยะห่างบนระนาบ n มิติของพิกัดแบบยูคลิด

$$d_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

อย่างไรก็ตามหากตัวแปรที่ใช้ในการบอกพิกัดนั้นมิได้สัมพันธ์กัน (Correlation) การหาระยะห่างด้วยสมการและวิธีดังกล่าวจะทำให้ได้ค่าที่ไม่สามารถใช้ได้ทางสถิติ เนื่องจากการใช้ค่าที่ไม่ได้มีการเปรียบเทียบบรรทัดฐาน ให้มีค่าเท่ากันนั้นเมื่อนำมาคำนวณร่วมกันจะทำให้ได้ข้อมูลที่มีความหมายกำกวมออกมา เช่น การคำนวณที่ไม่ได้อยู่ในรูปแบบของระยะห่างระหว่างแกน X Y Z หรือ การมีมากกว่า 3 ตัวแปรในการระบุพิกัด เป็นต้น เป็นเหตุให้ต้องมีการคำนวณระยะห่างระหว่างจุดใดๆ แบบอื่นเกิดขึ้นเพื่อใช้งานในด้านนี้โดยเฉพาะนั้นก็คือ MD

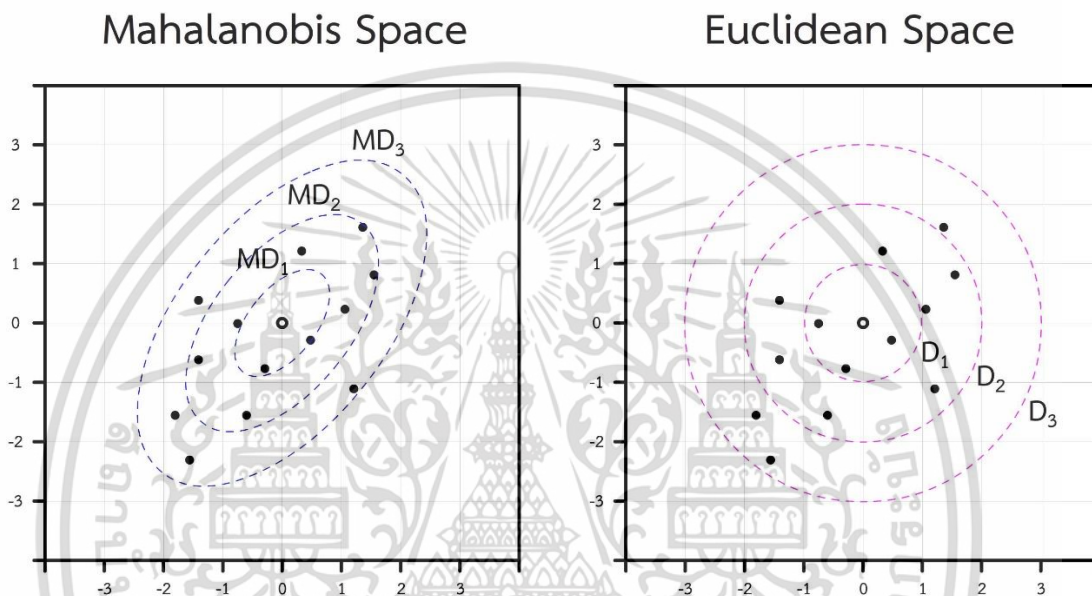
MD นั้นจะทำการอ้างอิงระยะห่างของทุกจุดบนระนาบเดียวกันโดยใช้จุดเซนทรอยด์ (Centroid) หรือจุดศูนย์กลางของมวล ซึ่งคำนวณได้จากค่าเฉลี่ยของทุกค่าบนแนวแกนนั้น ๆ การคำนวณนี้จะทำให้ได้ค่าที่ออกมาไม่มีหน่วย แต่ได้ค่าที่มีความสัมพันธ์กันในทุกแนวแกน โดยคำนวณได้จากสมการที่ (2.2)

$$MD_{xy} = \sqrt{\sum_{i=1}^n \frac{(x_i - y_i)^2}{S_i^2}} \quad (2.2)$$

โดยกำหนด x_i และ y_i คือตำแหน่งในแนวแกน x และ y ของมิติที่ i ตามลำดับ และให้ S คือค่าเบี่ยงเบนมาตรฐาน (Standard deviation) ในมิติที่ i ซึ่งจะเห็นได้ชัดว่าค่าที่ออกมานั้นจะมีความเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างกับการคำนวณแบบยูคลิดในทุกแนวแกนทำให้ค่าที่ออกมาแตกต่างกันอย่างสิ้นเชิงหากในแต่ละแนวแกนนั้นมี ปริภูมิ ที่ต่างกัน

ในปัจจุบันนั้น MD ใช้การเปลี่ยนแปลงมาตราส่วน (re-scale) ให้กับค่าต่าง ๆ ที่ต้องใช้ในการคำนวณค่าสัมประสิทธิ์ เช่น การคำนวณในเมทริกซ์ เป็นต้น ตัวอย่างการคำนวณ MD เมื่อเทียบกับการคำนวณในระบบของยูคลิดแบบ 2 มิติ จะแสดงในรูปที่ 2.4 จะเห็นได้ว่า หากจำนวนสมาชิกในระบบเท่ากันแต่ใช้การคำนวณคนละแบบนี้จะได้ผลลัพธ์ที่แตกต่างกัน



รูปที่ 2.4 เปรียบเทียบระยะห่าง MD และระยะห่างแบบยูคลิด

2.4 การแจกแจงแบบเอกรูปและแบบปกติ (Uniform and Normal Distribution)

การแจกแจงทางความน่าจะเป็น (probability distribution) คือการอธิบายโอกาสการเกิดขึ้นของกรณีทั้งหมดในการทดลอง ในรูปแบบของฟังก์ชันทางคณิตศาสตร์ที่ใช้ตัวแปรแซมเปิลสเปซ (Sample Space) และโอกาสของการเกิดเหตุการณ์นั้น ๆ ขึ้น เช่น ในการโยนเหรียญให้ออกหัวหรือก้อย ในที่นี้ ‘การโยนเหรียญ’ คือการทดลองที่ทำให้เกิดความเป็นไปได้ 2 อย่างเท่ากัน คือ การออกหัวมีความเป็นไปได้คือ 0.5 และการออกก้อยมีความเป็นไปได้คือ 0.5 เช่นเดียวกัน

ฟังก์ชันการสุ่มในระบบคอมพิวเตอร์ด้วยหน่วยประมวลผลหลักหรือ CPU นั้น ในทางหลักการแล้ว ไม่สามารถทำให้เกิดการสุ่มที่แท้จริงได้ (True RNG) กล่าวคือ ระบบคำนวณต่าง ๆ ที่ใช้ในปัจจุบันนั้นมีการคำนวณโดยใช้สัญญาณนาฬิกาเป็นตัวให้สัญญาณในการคำนวณ ผลของการคำนวณทุกอย่างที่เกิดขึ้นนั้น จะเป็นเช่นเดิมเสมอแม้กระทั่งการใช้เวลาเป็นส่วนหนึ่งของฟังก์ชัน ในปัจจุบันการสุ่มในคอมพิวเตอร์จึงนิยมสร้างลำดับของผลลัพธ์ที่จะออกมาก่อน (Pseudo Random

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sequence) แล้วจึงแสดงผลลัพธ์ออกมาตามลำดับ ซึ่งแต่ละลำดับจะตัวอ้างอิงของแต่ละลำดับที่เรียกว่า random seed ข้อเสียของการสุ่มด้วยกระบวนการนี้คือ หากรู้ซึ่ง seed ที่ระบุถึงลำดับดังกล่าวแล้ว จะทำให้สามารถคาดการณ์ลำดับต่อไปที่จะเป็นผลลัพธ์ของการสุ่มได้ จึงทำให้หลักการนี้เรียกว่าการสุ่มแบบเทียม (Pseudo Random) แต่ข้อดีก็คือสามารถกำหนดรูปแบบของการแจกแจงผลลัพธ์ออกมาได้ด้วย ฟังก์ชันการแจกแจงความน่าจะเป็นได้อย่างแม่นยำ

การแจกแจงแบบเอกรูปหรือ Uniform distribution คือรูปแบบของการแจกแจงที่ทุกผลลัพธ์ที่สามารถเป็นไปได้ระหว่างขอบ (Boundary) มีความเป็นไปได้เท่ากันทุกผลลัพธ์ ดังสมการการแจกแจงที่ (2.3) ซึ่งจะมีลักษณะเหมือนกับการโยนลูกเต๋าหรือการโยนเหรียญที่ไม่มีการถ่วง

$$P(i|a, b) = \frac{1}{b - a} \quad (2.3)$$

กำหนดให้ a คือขอบของค่าต่ำสุด (Lower bound) b คือ ขอบของค่าที่สูงที่สุด (Upper bound) และ i คือผลลัพธ์ของการสุ่ม ซึ่ง i อยู่ในช่วงของ [a,b] การสุ่มโดยใช้การแจกแจงแบบเอกรูปจะทำให้ลำดับที่ได้ออกมามีค่าเฉลี่ยและค่ามัธยฐานที่เท่ากันคือ $0.5(a+b)$ ตัวอย่างเช่น การสุ่มตัวเลข 1 ถึง 5 มา 5 ครั้ง อาจจะได้ผลลัพธ์เป็นลำดับ 2,3,5,1,4 หรือ 5,4,1,2,3 ซึ่งรับประกันว่าจะปรากฏความเป็นไปได้ทุกอย่างขึ้น

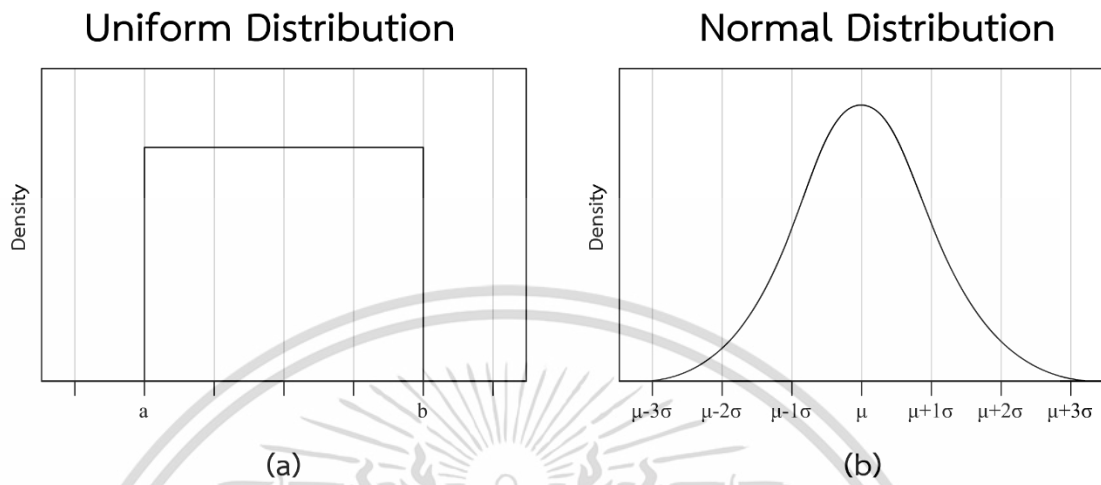
การแจกแจงแบบปกติ หรือ Normal distribution หรือเรียกได้อีกชื่อว่า การแจกแจงแบบเกาส์เซียน (Gaussian distribution) คือการแจกแจงที่สามารถกำหนดค่าเฉลี่ยหรือค่าที่คาดหวังของผลลัพธ์เพื่อให้ ลำดับที่ออกมามีแนวโน้มที่จะออกมาเป็นค่าใดมากที่สุด อีกทั้งยังสามารถกำหนดส่วนเบี่ยงเบนมาตรฐาน หรือ SD ของลำดับ เพื่อกำหนดขอบเขตของการสุ่มได้ ดังที่ได้อธิบายในสมการที่ (2.4)

$$P(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (2.4)$$

กำหนดให้ x เป็นผลลัพธ์ของการสุ่ม μ คือค่าเฉลี่ยหรือค่าที่คาดหวังดังกล่าวและ σ คือส่วนเบี่ยงเบนมาตรฐานดังกล่าว ซึ่งจะทำให้ลำดับที่ออกมามีค่าเฉลี่ย ค่ามัธยฐาน และค่าฐานนิยม ที่เท่ากับค่า μ

การสุ่มประเภทนี้มีความสำคัญทางสถิติเป็นอย่างมาก เนื่องจากเป็นการสุ่มส่วนใหญ่ที่เกิดขึ้นในธรรมชาติ การแจกแจงประเภทนี้มักใช้เป็นสมมติฐานในการหาค่าตอบของการสุ่มที่เกิดขึ้นโดย

ธรรมชาติ จะเห็นได้จากกราฟแสดงความน่าจะเป็นในรูปที่ 2.5a จะแสดงการสุ่มที่เกิดจากการแจกแจงแบบเอกรูปจาก a ถึง b และรูปที่ 2.5b แสดงความน่าจะเป็นของการแจกแจงแบบปกติ



รูปที่ 2.5 เปรียบเทียบความน่าจะเป็นระหว่างการกระจายแบบเอกรูปและการกระจายแบบปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 งานวิจัยที่เกี่ยวข้อง

3.1 วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization, PSO)

การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค คืออัลกอริทึมที่ถูกนำเสนอโดย James Kennedy และคณะ[4] เป็นการหาค่าที่เหมาะสมโดยการทำงานเป็นรายรอบ (Iteratively) เพื่อที่จะพัฒนาผลลัพธ์ให้ดียิ่งขึ้นไปเรื่อย ๆ ใช้แนวคิดที่ว่ามี อนุภาค หรือเรียกว่า particle จำนวนหนึ่งเคลื่อนที่ไปมาอยู่ที่ปริภูมิการค้นหา (search-space) โดยกำหนดให้อนุภาคแต่ละตัวมีความเร็วและตำแหน่งเพื่อใช้ในการเคลื่อนที่ อนุภาคจะเคลื่อนที่อย่างเป็นอิสระต่อกันโดยอาศัยแนวโน้มของจุดที่ดีที่สุดของแต่ละตัวหรือที่เรียกว่า Personal Best position (PBest) และจะมีจุดร่วมกันคือ การเคลื่อนที่ตามแนวโน้มของ อนุภาคที่ได้ค่าที่ดีที่สุด เรียกว่า Global Best position (GBest)

อัลกอริทึมนี้ได้รับแรงบันดาลใจมาจากการจำลองพฤติกรรมของสัตว์ที่อยู่กันเป็นกลุ่ม (Social behavior) ของฝูงปลาและฝูงนก ผู้วิจัยได้ทำการออกแบบจากแนวคิดดังกล่าวให้อยู่ในรูปแบบของกระบวนการประเภทเมตาฮีริสติก (Metaheuristic optimization algorithm) หมายความว่า PSO นั้นเป็นอัลกอริทึมที่ไม่ขึ้นกับปัญหา สามารถที่จะแก้ไขปัญหาล้วนใหญ่ได้อย่างมีประสิทธิภาพ ซึ่งเป็นข้อดีของอัลกอริทึมตระกูลนี้

กระบวนการทำงานของอัลกอริทึมจะเริ่มจากการกำหนดจำนวนของอนุภาคที่มีอยู่ในระบบ และทำการสุ่มตำแหน่งเริ่มต้นซึ่งก็คือ ตัวแปรการตัดสินใจ (decision vector) ของปัญหาขึ้นมา การเคลื่อนที่ของอนุภาคหรือการเปลี่ยนแปลงความเร็วและตำแหน่ง แต่ละตัวในอัลกอริทึมใน 1 รอบของการทำงาน สามารถอธิบายได้ดัง สมการ ที่ (3.1) และสมการที่ (3.2) นี้

$$v_{ij}^{t+1} = \omega v_{ij}^t + C_1 r_1^t (pbest_{ij} - x_{ij}^t) + C_2 r_2^t (gbest_j - x_{ij}^t) \quad (3.1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (3.2)$$

สมการที่ (3.1) อธิบายถึงการเปลี่ยนแปลงความเร็ว v ของอนุภาค x ตัวที่ i ในมิติที่ j ของรอบการทำงานที่ t ซึ่งจะส่งผลให้ความเร็วในรอบการทำงานต่อไปหรือในรอบที่ $t+1$ นั้นเกิดการเปลี่ยนแปลง ตัวแปร r_1 และ r_2 คือค่าที่ทำการสุ่มออกมาอย่างอิสระต่อกันโดยจะอยู่ในช่วง $[0,1]$ ตัว

แปร ω คือค่าแรงเฉื่อยของอนุภาคหรือ particle inertia มีค่าในช่วง $[0,1)$ ทำหน้าที่กำหนดอิทธิพลของความเร็วก่อนหน้าที่มีต่อความเร็วในครั้งต่อไปเปรียบเสมือนกับ น้ำหนักหาก ω มีค่ามากหมายถึง อนุภาคมีแนวโน้มที่จะเคลื่อนที่ไปยังทิศทางเดิมในครั้งถัดไป c_1 คือค่าสัมประสิทธิ์ การเรียนรู้แบบอิสระ (Individual-cognition) คือตัวแปรที่กำหนดอิทธิพลของ PBest ที่มีผลต่อการเปลี่ยนแปลงความเร็ว ส่วน c_2 นั้นทำหน้าที่คล้ายคลึงกับ c_1 แต่ใช้กับ GBest คือค่าสัมประสิทธิ์ การเรียนรู้จุ่มรวม (Social learning) คือตัวแปรที่กำหนดอิทธิพลของ GBest ที่มีผลต่อการเปลี่ยนแปลงความเร็ว

ต่อจากกระบวนการการเปลี่ยนแปลงความเร็วคือการแปลงตำแหน่งตามสมการที่ 3.2 โดยจะทำให้ได้ตำแหน่งใหม่ในรอบที่ $t+1$ ซึ่งจะถูกป้อนให้เป็นตัวแปรการตัดสินใจของฟังก์ชันของปัญหาต่อไป



รูปที่ 3.1 ตำแหน่งของอนุภาคในอัลกอริทึม PSO ในรุ่นที่แตกต่างกัน

รูปที่ 3.1 จำลองการทำงานของอัลกอริทึม PSO ในปัญหา 2 มิติโดยให้แกน x และ y แทนด้วยตัวแปรการตัดสินใจ ตัวที่ 1 และตัวที่ 2 ตามลำดับ กำหนดให้ เครื่องหมาย cross แทนด้วยอนุภาคทั้งหมดในรุ่นที่ 1 triangle แทนด้วย รุ่นที่ 60 จะเห็นได้ว่า เมื่ออัลกอริทึมทำงานในรอบที่มากขึ้นจะทำให้ข้อมูลไปกระจุกตัวอยู่ในพื้นที่ที่ใกล้เคียงกับ GBest มากขึ้น ทำให้จำกัดขอบเขตในการหาผลลัพธ์ได้ดียิ่งขึ้น

อย่างไรก็ตามในปัจจุบันมีงานวิจัยหลายชิ้นที่ชี้ให้เห็นว่าอัลกอริทึม PSO นั้นมีประสิทธิภาพที่ดีในการหาจุดต่ำสัมพัทธ์ เนื่องจากมีความละเอียดในการค้นหาพื้นที่รอบข้างเป็นอย่างมาก แต่ก็มีข้อเสียคือ ปัญหาการติดอยู่ในจุดต่ำสัมพัทธ์ (Trapping of local minima) เนื่องจากอนุภาคที่เข้าไปติดอยู่ในจุดดังกล่าวแล้วหากใช้ความเร็วเดิมต่อไปนั้น ก็ไม่อาจที่จะหลุดออกมาได้ จึงเป็นความท้าทายที่ต้องการการวิจัยและพัฒนาต่อไป ซึ่งอัลกอริทึม PSO นั้นได้เป็นต้นแบบให้กับอัลกอริทึมในรูปแบบ

Swarm intelligent อีกมากมาย เนื่องจากการนำไปพัฒนาต่อยอดจากปัญหาดังกล่าวและการพัฒนาประสิทธิภาพในด้านต่าง ๆ

3.2 วิธีหาค่าเหมาะสมที่สุดแบบฝูงหมาป่าเทา (Grey Wolf Optimizer, GWO)

อัลกอริทึม GWO ได้รับแรงบันดาลใจมาจากการล่าเหยื่อของฝูงหมาป่าสีเทาบริเวณขั้วโลกเหนือซึ่งเป็นสัตว์ที่อยู่ในจุดสูงสุดของห่วงโซ่อาหารในละแวกนั้น ในฝูงหมาป่านั้นจะแบ่งลำดับชั้นความสำคัญของหมาป่าแต่ละตัวในการทำหน้าที่ต่าง ๆ ถูกนำเสนอโดย Seyedali Mirjalili และคณะ [5]

GWO ได้ใช้แนวคิดของอัลกอริทึมวิวัฒนาการประยุกต์ให้เป็นอัลกอริทึมการหาค่าเหมาะสมที่สุด โดยจะแบ่งประชากรทั้งหมดในกลุ่มให้มีทั้งหมด 4 ลำดับตามหน้าที่และชนชั้น ประกอบไปด้วย

- หมาป่าอัลฟา (Alpha wolf : α) คือจ่าฝูงหรือประชากรที่มีความสำคัญที่สุด มีหน้าที่สั่งการประชากรตัวอื่น ให้ทำตาม
- หมาป่าเบต้า (Beta wolf : β) เป็นประชากรที่มีความสำคัญรองลงมาจาก อัลฟา มีหน้าที่ช่วยอัลฟาในการตัดสินใจทำสิ่งต่าง ๆ และช่วยอัลฟาในการตัดสินใจ
- หมาป่าโอเมก้า (Omega wolf : ω) คือประชากรที่อยู่ในชนชั้นต่ำสุดมีหน้าที่ทำตามที่จ่าฝูงสั่งการ เป็นประชากรกลุ่มที่ต้องทำการออกล่า
- หมาป่าเดลต้า (Delta wolf : δ) คือประชากรที่มีชนชั้นสูงกว่าโอเมก้าแต่ต่ำกว่าอัลฟา และเบต้า มีหน้าที่คล้ายคลึงกับเบต้า ในแง่ของการหาค่าที่เหมาะสมที่สุด

การทำงานของ GWO นั้นจะเลียนแบบการล่าเหยื่อของฝูงหมาป่า 1 ฝูงซึ่งจะแบ่งเป็น 4 ระยะคือ ล้อมเหยื่อ (Encircling prey) การล่า (Hunting) การโจมตีเหยื่อ (Attacking prey) และการหาเหยื่อตัวถัดไป (Search for prey)

ทุกรอบการทำงานมันจะมีการแบ่งชนชั้นนั้นเกิดขึ้นโดยการให้ ประชากรที่มีผลลัพธ์ที่ดีที่สุด (Fittest solution) เป็น หมาป่าอัลฟา และตามมาด้วย เบต้าและเดลต้าตามลำดับ ส่วนหมาป่าตัวอื่นๆ จะถูกจัดให้เป็น โอเมก้า

การล้อมเหยื่อ (Encircling prey) คือการหาตำแหน่งใหม่ของหมาป่าทุกตัว โดยภายใต้อิทธิพลของเหยื่อที่ทำการหามาเพื่อใช้เป็นเวกเตอร์การตัดสินใจ ของปัญหา สามารถอธิบายได้โดยสมการที่ (3.3) และสมการที่ (3.4) ดังต่อไปนี้

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \quad (3.3)$$

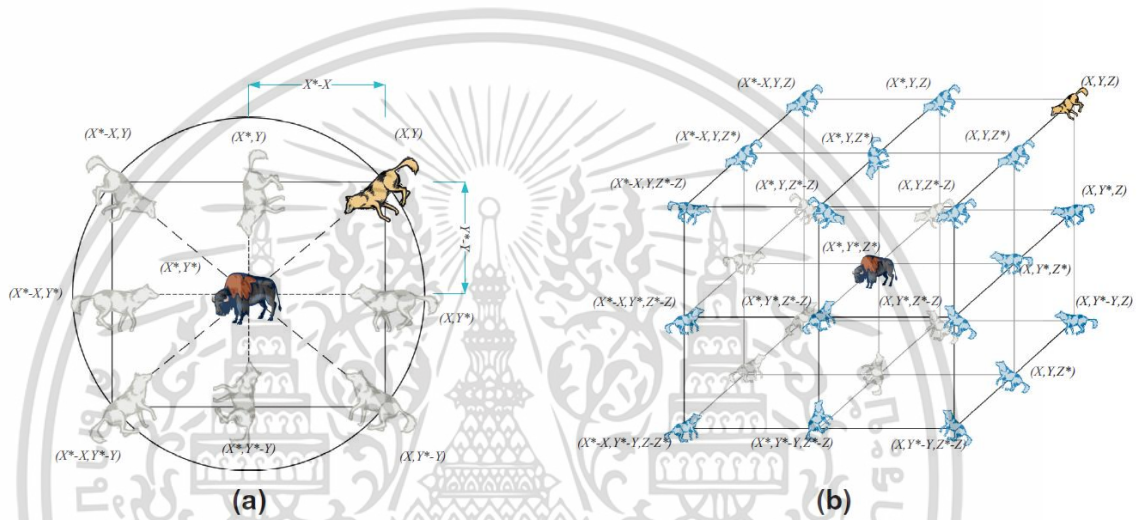
$$\vec{X}(t + 1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (3.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ X_p คือ เวกเตอร์ตำแหน่งของเหยื่อ $X(t)$ คือ ตำแหน่งของหมาป่าในรอบการทำงานที่ t A และ C คือค่าสัมประสิทธิ์ คำนวณได้จากสมการที่ (3.5)

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \vec{C} = 2 \cdot \vec{r}_2 \quad (3.5)$$

กำหนดให้ a คือตัวแปรที่มีค่าลดลงจาก 2 ไปยัง 0 แปรผันตามรอบการทำงาน r_1 และ r_2 คือตัวแปรที่ทำการสุ่มอย่างอิสระต่อกันโดยอยู่ในช่วง $[0,1]$ ทำให้ตำแหน่งใหม่ของหมาป่าที่จะเป็นไป ได้เกิดขึ้นตามรูปที่ 3.2 โดยรูปซ้าย (a) จะอยู่ในแบบของปัญหาแบบ 2 มิติ ส่วนรูปขวา (b) จะเป็น ปัญหาแบบ 3 มิติ



รูปที่ 3.2 ตำแหน่งที่เป็นไปได้ในระหว่างการล้อมเหยื่อของอัลกอริทึม GWO

การล่าเหยื่อ (Hunting) คือการเคลื่อนที่ของหมาป่าในกรณีที่ไม่มีเหยื่อที่แน่นอน หมาป่าทุกตัวจะเคลื่อนที่ตาม หมาป่าจำฝูง 3 ตัวหลักตามสมการดังต่อไปนี้

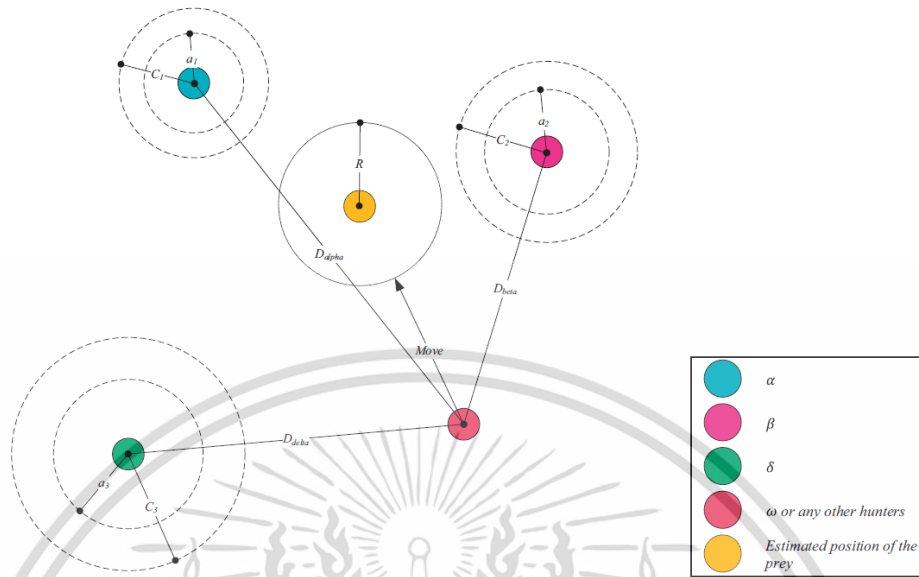
$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (3.6)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A} \cdot \vec{D}_\alpha, \vec{X}_2 = \vec{X}_\beta - \vec{A} \cdot \vec{D}_\beta, \vec{X}_3 = \vec{X}_\delta - \vec{A} \cdot \vec{D}_\delta \quad (3.7)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.8)$$

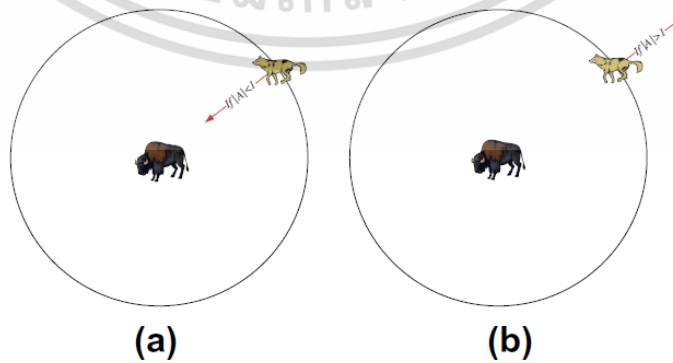
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงให้เห็นถึงการเคลื่อนที่ของหมาป่าตัวอื่นตามหมาป่าล่าฝูง โดยใช้สมการที่ (3.8) ซึ่งคำนวณมาจากตำแหน่งของประชากรที่ดีที่สุด 3 ตัวแรก



รูปที่ 3.3 ตำแหน่งที่เป็นไปได้ในระหว่างการล่าเหยื่อของอัลกอริทึม GWO

การโจมตีเหยื่อ (Attacking prey) และการหาเหยื่อตัวถัดไป (Search for prey) เมื่อจำนวนรอบการทำงานเข้าใกล้จุดสิ้นสุด ตัวแปร A ในสมการที่ 3.5 นั้นจะมีช่วงที่แคบลงเนื่องจากการลดลงของตัวแปร a ซึ่งจะทำให้การทำงานของอัลกอริทึมนั้นมีพฤติกรรมที่เปลี่ยนไปเนื่องจาก หากตัวแปร A ที่สุ่มได้นั้นมีค่าน้อยกว่า 1 หมาป่าตัวดังกล่าวจะเคลื่อนที่เข้าใกล้เหยื่อมากขึ้นเกิดเป็นการโจมตีเหยื่อ และในทางเดียวกัน หากตัวแปร A นั้นมีค่ามากกว่า 1 จะทำให้หมาป่าตัวดังกล่าวออกห่างจากเหยื่อมากขึ้นเป็นการหาเหยื่อตัวถัดไป ดังรูปที่ 3.4



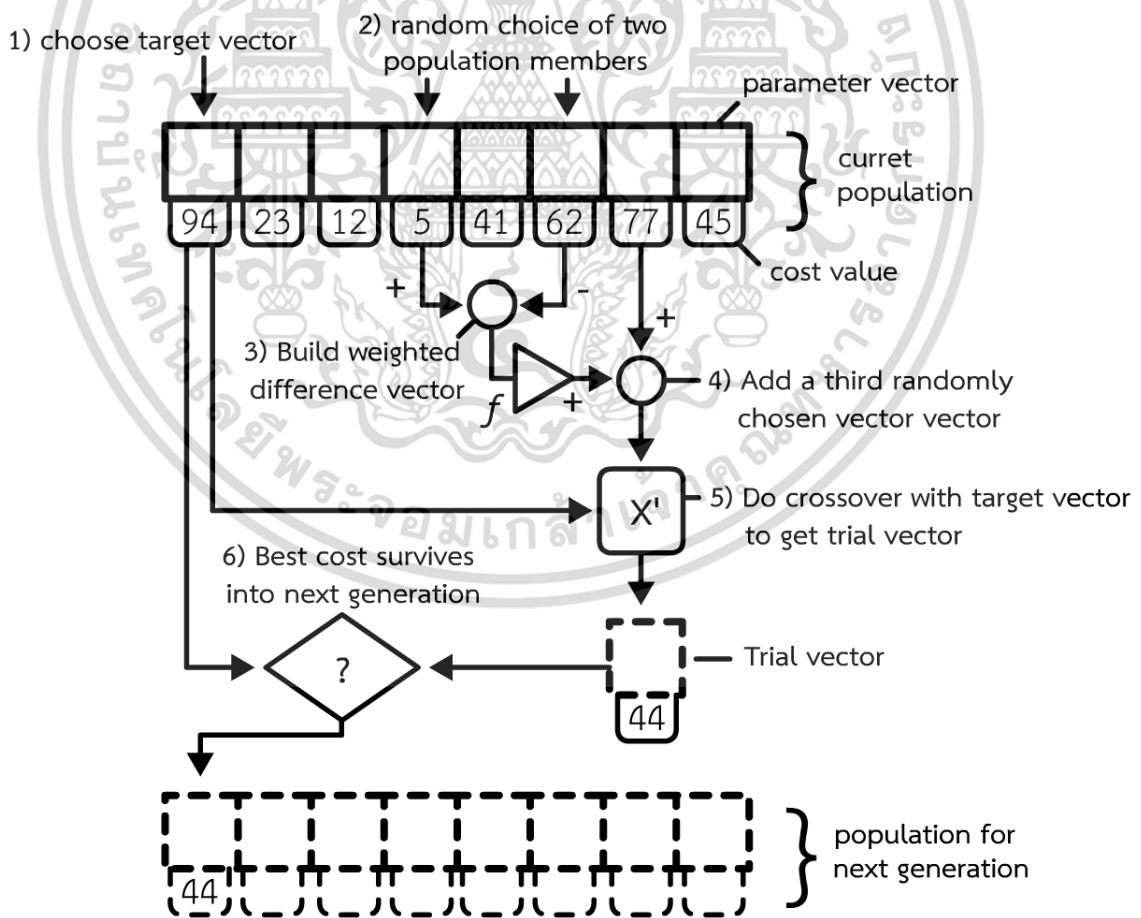
รูปที่ 3.4 ตัวแปร A ที่กำหนดการโจมตีเหยื่อและการหาเหยื่อตัวถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วิธีหาค่าเหมาะสมที่สุดแบบวิธีวิวัฒนาการโดยใช้ผลต่าง (Differential Evolution, DE)

อัลกอริทึม DE เป็นการหาค่าที่เหมาะสมที่สุดที่จัดอยู่ในหมวดหมู่ของอัลกอริทึมวิวัฒนาการ และเป็นอัลกอริทึมไม่กึ่งประเภทในตระกูลเมตาฮีริสติก (Metaheuristic) ที่นอกจากจะใช้ผลลัพธ์ในการตัดสินใจกระบวนการการทำงานแล้วยังมีการประเมินค่าคุณภาพ (Measure of quality) ของผลลัพธ์เพื่อใช้ในการปรับเปลี่ยนกลยุทธ์ซึ่งทำให้สามารถรองรับปัญหาที่หลากหลายแม้กระทั่งปัญหาแบบหลายวัตถุประสงค์ (Multi-objective optimization problem) ถูกนำเสนอโดย Rainer Storn และคณะ[6]

DE นั้นใช้แนวคิดของอัลกอริทึมวิวัฒนาการเป็นหลัก โดยจะใช้หลักการของประชากรหรือเรียกอีกชื่อว่า search agents และผลลัพธ์ที่เรียกว่า candidate solution ประชากรแต่ละตัวนั้นจะมีหน้าที่หาผลลัพธ์ที่ดีที่สุดทั่วทั้งปริภูมิ และจะใช้ฟังก์ชันคณิตศาสตร์พื้นฐานในการรวมประชากรจำนวนหนึ่งซึ่งขึ้นอยู่กับกลยุทธ์ที่เลือกใช้ ทำให้ให้เกิดเป็นประชากรตัวใหม่ที่คาดหวังว่าผลลัพธ์นั้นจะดีขึ้น หากผลลัพธ์ใหม่ดังกล่าวดีขึ้น อัลกอริทึมรับประชากรตัวใหม่นั้นเข้ามาอยู่ในระบบ หากจะไม่เพิกเฉยต่อคำตอบใหม่นั้น



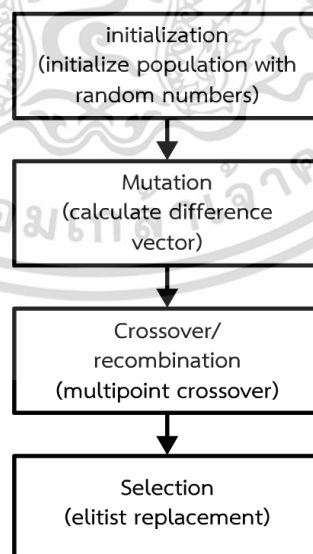
รูปที่ 3.5 ขั้นตอนการสร้างประชากรตัวใหม่ของอัลกอริทึมวิวัฒนาการโดยใช้ผลต่าง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 อธิบายถึงขั้นตอนของการทำงานของอัลกอริทึม DE โดยเริ่มจาก 1) เลือกประชากรในแต่ละรอบซึ่งโดยปกติจะเลือกทุกประชากร 2) ทำการสุ่มประชากรข้างเคียงมาจำนวนหนึ่ง (ขึ้นอยู่กับกลุ่ของที่ใช้) 3,4,5) ใช้ฟังก์ชันทางคณิตศาสตร์เพื่อสร้างผลลัพธ์ใหม่ (candidate solution) 6) ประเมินว่าผลลัพธ์ใหม่นั้นควรจะอยู่ในกลุ่มประชากรใหม่หรือไม่ จากกระบวนการทำงานของ DE จะเห็นได้ว่า กลยุทธ์ที่เลือกใช้นั้นมีความสำคัญเป็นอย่างมากโดย กลยุทธ์นั้นจะเป็นตัวกำหนดวิธีการเลือกประชากร และจำนวนของประชากร ที่จะนำมาคำนวณผลลัพธ์

กลยุทธ์หลัก ๆ ที่ใช้ในปัจจุบันจะเป็นการผสมผสานระหว่างคุณสมบัติ 3 อย่างคือ[7]

- Parent หรือประชากรต้นแบบที่จะนำมาเป็นต้นแบบในการสร้างประชากรรุ่นถัดไป เช่น ใช้ประชากรที่ดีที่สุด (Best) สุ่มประชากรตัวใหม่ (Rand) และ ใช้ค่าของประชากรสุ่ม 1 ตัว ร่วมกับตัวที่ดีที่สุดอีก 1 ตัว (Best-to rand)
- จำนวนคู่ของประชากรที่จะนำมาคำนวณเพื่อสร้างประชากรใหม่
- ชนิดของการ crossover โดยปกติจะมีให้เลือก 2 ประเภทคือ Binomial crossover (bin) และ exponential crossover (exp)

ตัวอย่างเช่น DE/Rand/1/exp คือการใช้ประชากรที่สุ่มมา 1 ตัวมาผสมกับการ exponential crossover ทำให้เกิดผลลัพธ์ใหม่ หลังจากสร้างประชากรรุ่นใหม่แล้วจะทำการคัดเลือกประชากร (Selection) ทั้งหมดในการใช้เป็นประชากรตั้งต้นในรอบต่อไป กระบวนการดังกล่าวจะทำงานจนถึงรอบการทำงานที่กำหนดตาม ผังการทำงาน (Flow chart) ในรูปที่ 3.6 จนกระทั่งถึงเงื่อนไขรอบการทำงานหรือจนกว่าจะเจอคำตอบ



รูปที่ 3.6 ผังการทำงานของอัลกอริทึมวิวัฒนาการโดยใช้ผลต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วิธีหาค่าเหมาะสมที่สุดแบบรังผึ้งสังเคราะห์ (Artificial bee colony, ABC)

Dervis Karaboga และคณะนำเสนอ อัลกอริทึมการหาค่าที่เหมาะสมที่สุดแบบรังผึ้งสังเคราะห์ ซึ่งได้รับแรงบันดาลใจมาจาก การที่ผึ้งที่อยู่ในรังช่วยกันหาอาหารโดยแลกเปลี่ยนข้อมูลของที่อยู่ของแหล่งอาหาร (Food source) ที่ดีที่สุด เปรียบเสมือนกับการหาจุดที่ดีที่สุดของโจทย์ปัญหา [8]

การทำงานของอัลกอริทึม จะสมมุติว่ามีผึ้งหุ่นยนต์ซึ่งก็คือหน่วยการทำงานย่อย (Search agent) อยู่จำนวนหนึ่ง โดยจะแบ่งผึ้งเป็น 3 กลุ่มตัวกันคือ ผึ้งงาน ผึ้งเฝ้าดู และ ผึ้งสำรวจ เริ่มแรกจะแบ่งครึ่งประชากรทั้งหมดให้มี 2 กลุ่มเท่ากันและกำหนดให้เป็นผึ้งงานและผึ้งเฝ้าดู จากนั้นจะแบ่งหน้าที่และการทำงานของผึ้งแต่ละกลุ่มดังต่อไปนี้

- ผึ้งงาน (Employed Bees) มีหน้าที่รายงานบริเวณที่คิดว่าเป็นแหล่งอาหารที่ดีที่สุด และค้นหาจุดที่ดีที่สุดโดยใช้การวิวัฒนาการแบบผ่าเหล่า (Mutate) ในการค้นหาตำแหน่งถัดไป ตามสมการที่ 3.9

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \quad (3.9)$$

ให้ v_{ij} คือตำแหน่งใหม่ที่จะทำการค้นหาในมิติที่ j ของผึ้งงานหมายเลข i จากตำแหน่งเดิมคือ x_{ij} โดย φ คือตัวแปรที่ทำการสุ่มแบบอิสระต่อ i และ j อยู่ในช่วง $[-1,1]$ ในแต่ละรอบการทำงานนั้นผึ้งที่ตัวซึ่งระบุโดยหมายเลข i จะทำการสุ่ม j ซึ่งเป็นโครโมโซมที่จะทำการผ่าเหล่า โดยการสุ่มผึ้งงาน หมายเลข k มาทำการคำนวณตามสมการที่ (3.9)

- ผึ้งเฝ้าดู (Onlooker Bees) มีหน้าที่รับตำแหน่งจากผึ้งงานและค้นหาบริเวณรอบ ๆ ของแหล่งอาหารที่ผึ้งงานได้หาไว้ให้ โดยจะทำการสุ่มแหล่งอาหารดังกล่าวโดยมีความน่าจะเป็นของแต่ละแหล่งอาหารตามสมการที่ 3.10 ซึ่งจะแปรผันตรงตามค่าผลลัพธ์ที่ดีที่สุด ณ ขณะนั้น

$$p_i = \frac{fit_i}{\sum_n^{SN} fit_n} \quad (3.10)$$

ให้ P_i คือโอกาสที่ผึ้งเฝ้าดูแต่ละตัวจะไปค้นหาแหล่งอาหารที่ i fit_i และ fit_n คือค่าผลลัพธ์ (Fitness value) ของแหล่งอาหารของผึ้งงานตัวที่ i และ n ตามลำดับ SN คือจำนวนของผึ้งงาน เมื่อผึ้งงานหรือผึ้งเฝ้าดูไปค้นหาที่แหล่งอาหารใดๆ แหล่งอาหารนั้นถูกบันทึกลงไปหน่วยความจำของผึ้งงานว่าได้ทำการค้นหาไปแล้วกี่ครั้งในแหล่งอาหารดังกล่าว และเมื่อครบจำนวนครั้งของการค้นหาที่ไม่ทำให้เกิดผลที่ดีขึ้น (maximum cycle number : MCN) จะถือว่าแหล่งอาหารนั้นถูกค้นหาจนทั่ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้ว แหล่งอาหารดังกล่าวจะถูกทิ้งโดยผึ้งงานที่ดูแล และผึ้งงานตัวดังกล่าวจะเปลี่ยนมาเป็นผึ้งสำรวจแทน

- ผึ้งสำรวจ (Scouts Bees) เมื่อแหล่งอาหารที่ผึ้งงานหรือผึ้งเฝ้าดู ตัวใด ๆ นั้นถูกค้นหาจนทั่วพื้นที่แล้ว ผึ้งตัวนั้นจะกลายเป็นผึ้งสำรวจเพื่อหาแหล่งอาหารใหม่ต่อไป การหาแหล่งอาหารใหม่นั้นจะใช้การสุ่มทุกมิติการค้นหาเช่นเดียวกับการสร้างข้อมูลตั้งต้นของอัลกอริทึม การทำงานนั้นจะแบ่งระยะการทำงาน (phase) เป็น 3 ระยะตามหน้าที่ ซึ่งในเบื้องต้นจะกำหนดให้ประชากรทุกตัวเป็นผึ้งงานทั้งหมด จากนั้นจะให้ประชากรทุกตัวทำงานเป็นผึ้งเฝ้าดู เมื่อจบรอบการทำงานทุกรอบจะทำการประเมินการทิ้งแหล่งอาหารและทำหน้าที่ในส่วนของผึ้งสำรวจทันที ทำให้จำนวนของผึ้งเฝ้าดูเท่ากับจำนวนประชากรและเท่ากับจำนวนของผึ้งงานรวมกับผึ้งสำรวจในทุกรอบ การทำงานทั้งหมดจะกระทำจนกว่าจะถึงรอบการทำงานที่กำหนดไว้หรือเมื่อหาคำตอบที่ดีที่สุดพบตาม pseudo code ในรูปที่ 3.7

```

1 initialize
2 Repeat :
3   Place the employed bees on the food source in memory
4   Place the onlooker bees on the food source in memory
5   Send the scouts to the search area for discovering
   new food source
6 Until (requirement are met)
  
```

รูปที่ 3.7 pseudo code ของอัลกอริทึม artificial bee colony

3.5 การวิเคราะห์และการเปรียบเทียบประสิทธิภาพของอัลกอริทึมในรูปแบบของ CEC

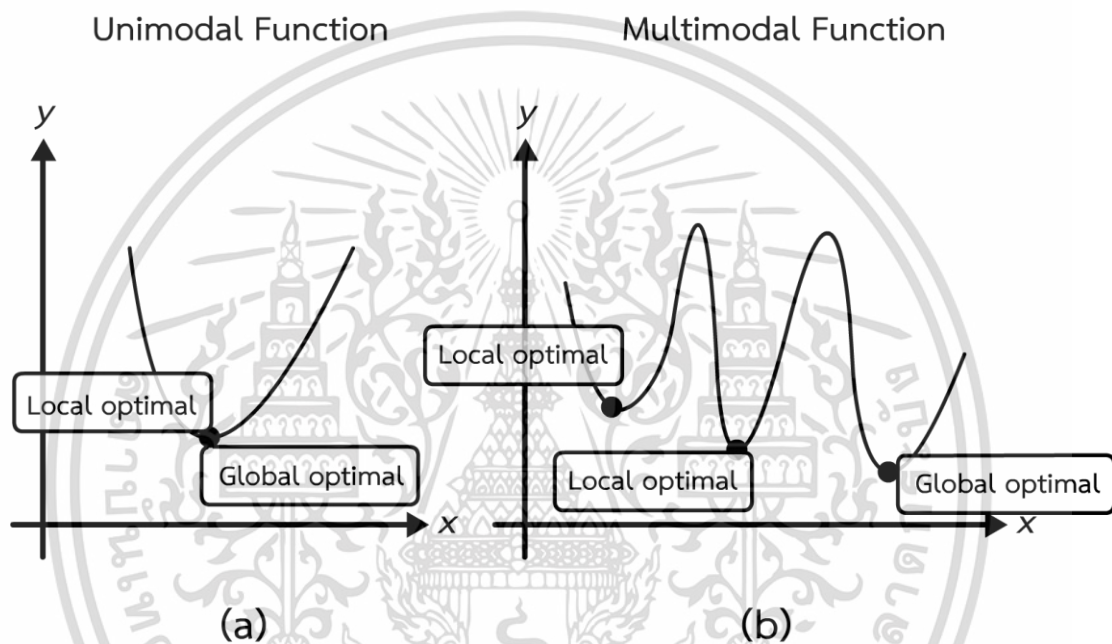
การวิจัยและพัฒนาอัลกอริทึมการหาค่าเหมาะสมที่สุดนั้น มีจุดมุ่งหมายเพื่อสร้างอัลกอริทึมที่มีประสิทธิภาพดีที่สุด และได้ทำการพัฒนามาจากผู้วิจัยหลากหลายคณะและหลากหลายแนวคิด การที่จะประเมินประสิทธิภาพของแต่ละอัลกอริทึมด้วยความเท่าเทียม (Fairness) นั้นจึงเป็นเรื่องสำคัญที่นักวิจัยควรให้ความสำคัญ จากเหตุผลดังกล่าว งานสัมมนาทางวิชาการ IEEE Congress on Evolutionary Computation (CEC) ซึ่งเป็นงานสัมมนาที่มีเนื้อหาเกี่ยวกับอัลกอริทึมการหาค่าเหมาะสมที่สุดโดยตรง จึงได้ออกกติกาและข้อกำหนดในการประเมินประสิทธิภาพอัลกอริทึมออกมาเพื่อเป็นแนวทางในการวิจัยด้านนี้ ในรูปแบบของปัญหาการหาค่าเหมาะสมที่สุด

หนึ่งในประเภทของปัญหาที่ CEC ให้ความสำคัญนั้นก็คือ Single Objective Real-Parameter Numerical Optimization เนื่องจากเป็นพื้นฐานของการหาค่าเหมาะสมที่สุดและสามารถพัฒนาให้ใช้กับงานอื่นๆ CEC จะทำการกำหนดปัญหาดังกล่าวออกมาเรื่อย ๆ ในทุก ๆ 3 – 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปี เช่น CEC2014 [9] CEC2013 [10] และ CEC 2005 [11] เป็นต้น ปัญหาจะแบ่งออกเป็น 2 หมวดหมู่หลัก ๆ คือ

- ปัญหาแบบฐานนิยมเดียว (Unimodal function) คือฟังก์ชันที่มีค่าต่ำสุดสัมพัทธ์ (Local minimum) เพียงจุดเดียวนั้นก็คือเป็นค่าต่ำสุดสัมบูรณ์ (Global minimum) คือคำตอบที่ดีที่สุดของฟังก์ชันนั่นเอง แสดงในรูปที่ 3.8a
- ปัญหาแบบพหุฐานนิยม (Multimodal function) คือฟังก์ชันที่มีค่าต่ำสุดสัมพัทธ์หลายจุดทั่วทั้งปริภูมิการค้นหา (Search-space) แสดงในรูปที่ 3.8b



รูปที่ 3.8 ปัญหาแบบแบบฐานนิยมเดียวและพหุฐานนิยมใน 1 มิติ

นอกจากนี้เพื่อความท้าทายในการแก้ปัญหา ยังมีฟังก์ชันผสม (Hybrid function) ที่เกิดจากการรวมกันระหว่างฟังก์ชันมาตรฐาน มากกว่า 1 ฟังก์ชัน และ ฟังก์ชันคอมโพสิต (Composite function) ที่เกิดจากเรียกกันเองของสองฟังก์ชันขึ้นไปอีกด้วย

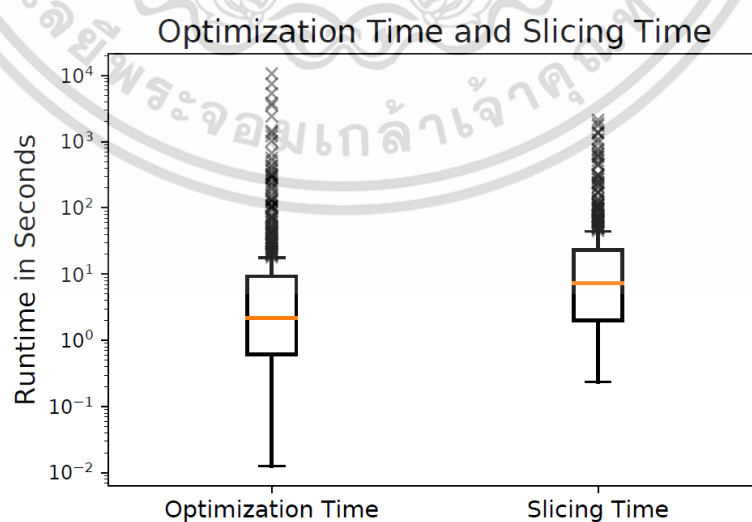
เพื่อความเท่าเทียมในการทดสอบ CEC ได้ใช้กำหนดรอบของการคำนวณคำตอบ (Function evaluation count) สูงสุดที่จะให้ใช้ในการคำนวณโดยจะแปรผันตาม จำนวนของมิติของปัญหา

3.6 การผสมผสานกลศาสตร์การเคลื่อนไหวเพื่อปรับปรุงเส้นทางของเครื่องจักรแบบหัวฉีด (Incorporating Kinematic Properties into Fused Deposition Toolpath Optimization)

Samuel และคณะ ได้พัฒนาเครื่องพิมพ์ 3 มิติที่สามารถทำงานในการเคลื่อนไหวในแนวแกน Z ขึ้นมา เรียกว่า “3D-aware printer” ด้วยแนวความคิดที่จะพัฒนาให้การพิมพ์วัตถุ หลายชั้นที่แยกจากกัน “spatially separated” ได้เร็วกว่าเครื่องพิมพ์ 3 มิติแบบทั่วไป แต่เนื่องจาก การพิมพ์วัตถุจากโมเดล 3 มิติ จะต้องผ่านขั้นตอนที่เปลี่ยนโมเดล 3 มิติให้กลายเป็นคำสั่งที่เครื่องปริ้นทำงาน (Slicing) ซึ่งเป็นส่วนสำคัญในการกำหนดเวลาการทำงานของการพิมพ์ทั้งหมด แต่เนื่องจากวิธีการดังกล่าวในปัจจุบันใช้การแยกโมเดลในแนวแกน Z ให้เหลือเป็นระนาบแบนที่อยู่ในแกน XY ทั้งหมด แล้วจึงคำนวณหาเส้นทางของเครื่องจักร (toolpath) ทำให้ เครื่องพิมพ์ที่ Samuel คิดค้นขึ้นมา ไม่ได้ลดเวลาในการปริ้นแต่อย่างใด

Samuel และคณะจึงได้คิดค้นอัลกอริทึมที่ทำให้การพิมพ์วัตถุแต่ละชั้นนั้นสามารถพิมพ์แยกกันได้แต่ปัญหาของเขาคือ การพิมพ์วัตถุที่ละชั้นนั้นบางครั้งก็ไม่สามารถทำงานได้เนื่องจากข้อจำกัดของเครื่องจักรและบางครั้งการพิมพ์แยกชั้นทำให้ใช้เวลาเพิ่มขึ้น จึงได้นำอัลกอริทึม การค้นหาเฉพาะที่ (Local Search) เข้ามาช่วยเป็นตัวตัดสินใจการกระทำในแต่ละเลเยอร์ของการ slicing เป็นบทความทางวิชาการ [12]

โดยได้จำลองวิธีดังกล่าวกับโมเดลมากกว่า 500 รูปแบบและผลปรากฏว่า มากกว่า 30% ของโมเดลนั้นได้เวลาที่ดีขึ้นมากกว่า 3.2% ซึ่งเป็นที่น่าพอใจเนื่องจากการพิมพ์ 3 มิตินั้นต้องใช้เวลาอย่างมาก การเปลี่ยนแปลงเพียงเล็กน้อยทำให้เวลาในหน่วยนาที่ลดลงอย่างมาก รูปที่ 3.9 แสดงให้เห็นเวลาจากอัลกอริทึมดังกล่าวเปรียบเทียบกับ การ slicing ที่ใช้ทั่วไป



รูปที่ 3.9 กราฟเทียนแสดงสถิติระหว่างอัลกอริทึม LS และการ slicing ทั่วไป

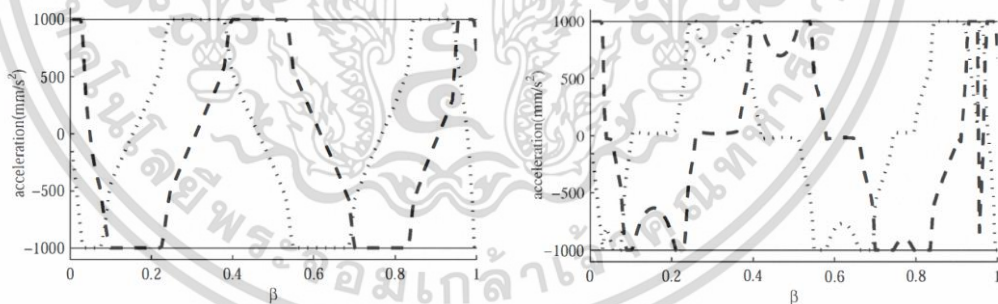
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การหาเวลาที่ที่ดีที่สุดของการกัดชิ้นงานโดยมีความเร่งเป็นเงื่อนไข (Minimum Time Feed-rate Optimization along Predefined Toolpath with Acceleration Constraints on Each Axis)

การผลิตชิ้นงานด้วย เครื่องจักรกลอัตโนมัติที่ทำงานได้ด้วยระบบคอมพิวเตอร์ หรือ CNC (Computer Numerical Control) นั้นจำเป็นต้องสร้างเส้นทางของหัวกัด (Toolpath) ซึ่งการสร้างเส้นทางดังกล่าวจำเป็นต้องใช้ข้อจำกัดของเครื่องจักรเป็นหนึ่งในพารามิเตอร์ที่ช่วยคำนวณเส้นทางที่ใช้ได้จริง ZHANG และคณะ [13] ได้คิดค้นการสร้างเส้นทางดังกล่าวโดยสามารถกำหนดขอบเขตของอัตราเร่งที่อนุญาตให้เครื่องจักรสามารถใช้งานได้ โดยใช้วิธีการ sequential quadratic programming (SQP) ซึ่งเป็นการแก้ไขปัญหามีข้อจำกัด (Constraints optimization problem)

ความท้าทายของปัญหาดังกล่าวคือการที่ ZHANG และคณะได้เปลี่ยนแปลงเส้นทางที่เครื่องจักรต้องเคลื่อนที่ให้กลายเป็นปัญหาเชิงการหาค่าที่เหมาะสมที่สุดแบบมีข้อจำกัด ทำให้สามารถใช้การแก้ไขปัญหเชิงตัวเลขซึ่งจะให้คำตอบที่ดีที่สุด (Best solution) อยู่เสมอ เป็นที่มาของเส้นทางที่ใช้เวลาและความเร็วได้คุ่มค่าที่สุดหรือ (Minimum Time Feed-rate)

รูปที่ 3.10 คือกราฟเปรียบเทียบระหว่างเส้นทางที่ไม่ใช้และใช้ความเร่งในการคำนวณ จุด (dot) และ ชีด (dash) คือความเร่งในแนวแกน x และ y ตามลำดับ ซึ่งในรูปช้ายนั้นคือการไม่ใช้ความเร่งในการคำนวณจะทำให้ความเร็วสูงสุดที่เครื่องจักรใช้อาจเกินขีดจำกัดของเครื่องจักรได้

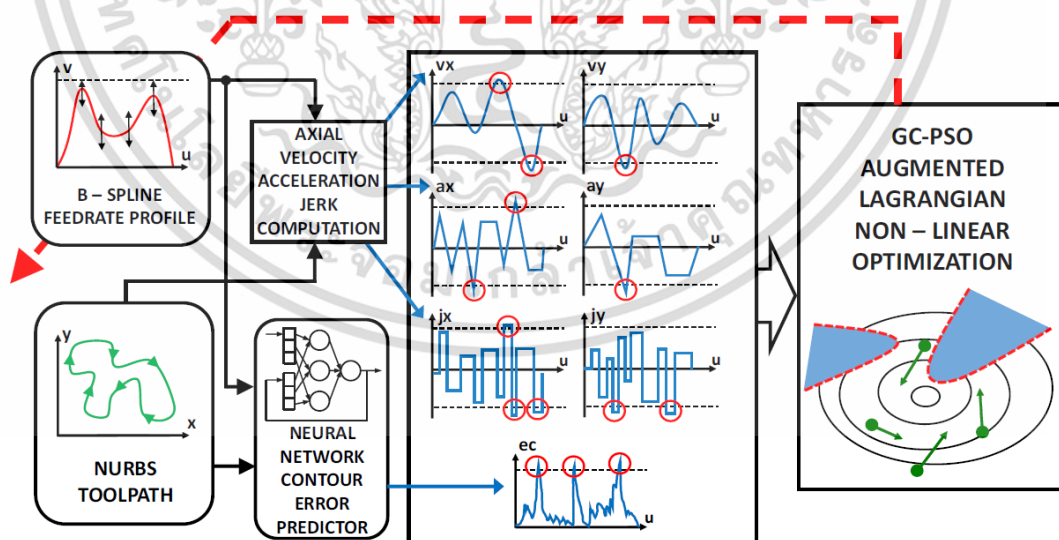


รูปที่ 3.10 กราฟเปรียบเทียบความเร่งระหว่างการคิดเส้นทางแบบไม่กำหนดความเร่ง (รูปช้าย) และแบบกำหนดความเร่ง (รูปขวา)

3.8 การใช้อัลกอริทึม PSO เพื่อกำหนดอัตราการกินวัสดุที่เหมาะสมที่สุดบนพื้นฐานของการเขียนแบบ NURBS (PSO based feedrate optimization with contour error constraints for NURBS toolpaths)

เส้นโค้ง NURBS (Non-Uniform Rational B-Spline) คือ เส้นโค้งที่เขียนแทนด้วยสมการ B-Spline หลาย ๆ จุดซึ่งอยู่ในรูปของอนุพันธ์ที่มีพจน์เท่ากับจุดควบคุมของค่าที่ต้องการประมาณ ออกมานิยมใช้แทนจุดเป้าหมายของเครื่องจักรประเภท CNC ที่เดิมใช้การ sampling จุดเป้าหมายทั้งหมดตามความละเอียด ซึ่งหากใช้เทคนิคดังกล่าว จะเปลี่ยนจากการกำหนดจุดที่แน่นอนจำนวนมหาศาลให้เหลือเพียงสมการที่ใช้แทนจุด XY บนระนาบ สามารถทำให้ระบุจุดเป้าหมายได้ในทุกช่วงของจำนวนจริงจากการหาค่าในเส้นโค้ง NURBS

การใช้เครื่องจักรประเภท CNC นั้นจะต้องทำการกำหนดอัตราเร็วที่ต้องการให้เครื่องมือ (Endmill tool) กินผิววัสดุเรียกว่า feedrate ซึ่งโดยปกติจะใช้การกำหนดเป็นจุดหรือค่าคงที่และทำการ sampling เมื่อนำไปใช้งาน แต่หากรูปแบบของชิ้นงานนั้นใช้เส้นโค้ง NURBS การกำหนดความเร็วหรือ feedrate ในรูปแบบธรรมดาอาจไม่ให้ค่าที่ละเอียดเพียงพอกับแบบที่เขียนมาได้ Krystian และคณะจึงได้เปลี่ยนจากการตั้งค่า feedrate จากเดิมเปลี่ยนเป็นการใช้เส้นโค้ง B-Spline ซึ่งสามารถแทนด้วยฟังก์ชันของความเร็ว และใช้อัลกอริทึม PSO นั้นทำให้ feedrate ที่ออกมามีผลทำให้ ความเร็ว ความเร่ง และ อัตราการติดกลับ (jerk) เป็นไปตามข้อกำหนด (Constrain) ที่สัมพันธ์กับเครื่องจักรนั้น ๆ [14]



รูปที่ 3.11 ตัวอย่างความสัมพันธ์ในรูปแบบของแผนภูมิต้นไม้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การเพิ่มประสิทธิภาพให้กับอัลกอริทึมการหาค่าที่เหมาะสมที่สุด

4.1 บทนำ

ในบทนี้จะนำเสนองานวิจัยที่ได้ทำการศึกษาและพัฒนาความสามารถของอัลกอริทึมการหาค่าที่เหมาะสมที่สุด โดยยึดหลักของอัลกอริทึม PSO และพัฒนาคุณสมบัติในด้านที่อัลกอริทึม PSO ยังด้อยเมื่อเทียบกับอัลกอริทึมสมัยใหม่ จึงสรุปได้ออกมาเป็นงานวิจัย 2 งาน นั่นก็คือ การพัฒนาในด้านของเวลาการประมวลผลโดยใช้ความสามารถของหน่วยประมวลผล (CPU) ในปัจจุบันซึ่งมีหลายหน่วยประมวลผลทางตรรกะ (Logical processing) ทำให้สามารถประมวลผลแบบขนาน (Parallel computation) และทำให้ลดเวลาในการหาค่าตอบของการหาค่าที่เหมาะสมที่สุดได้ ในชื่อของ การเพิ่มประสิทธิภาพด้านเวลาของการหาค่าเหมาะสมที่สุดแบบแบ่งกลุ่มอนุภาคโดยใช้การประมวลผลแบบขนาน (A time improvement PSO base algorithm using multithread programming) [15] และอีกงานวิจัยหนึ่งโดยมีจุดประสงค์เน้นที่การทำให้ผลลัพธ์นั้นออกมาเหมือนเดิมทุกครั้งซึ่งจะทำให้อัลกอริทึมนั้น พ้นขีดจำกัดของการทำงานแบบสุ่ม (Stochastic) มีผลให้การประเมินการทำงานทางสถิตินั้นออกมาดีกว่าต้นฉบับมาก ในชื่อของ การประมวลผลค่าที่เหมาะสมที่สุดแบบไม่สุ่มเพื่อคำตอบที่คงที่ (Hard Deterministic Particle Swarm Optimisation for Certain Result Solution) [16]

4.2 การเพิ่มประสิทธิภาพด้านเวลาของการหาค่าเหมาะสมที่สุดแบบแบ่งกลุ่มอนุภาคโดยใช้การประมวลผลแบบขนาน (A time improvement PSO base algorithm using multithread programming)

โดยปกติอัลกอริทึมจำพวก PSO จะสร้างอนุภาค (หน่วยย่อยในการประมวลผล) ขึ้นมาจำนวนหนึ่งเพื่อหาค่าตอบของปัญหาที่ดีที่สุด ในการใช้งานปกติแต่ละอนุภาคจะมีการประมวลผลและมีทรัพยากรเป็นของตัวเองอนุภาคเองและประมวลผลทีละตัวแบบโดยใช้เพียง 1 หน่วยประมวลผลหรือเรียกว่าการประมวลผลแบบเป็นลำดับ (Sequential processing) เป็นเหตุให้ต้องใช้เวลาในการประมวลผลเพิ่มขึ้นเมื่อมีจำนวนของอนุภาคในระบบเพิ่มมากขึ้น ส่งผลให้ระบบทำงานช้าลงเมื่อต้องคำนวณในระบบที่มีอนุภาคจำนวนมากและมีปัญหาเรื่องการประเมินประสิทธิภาพของอัลกอริทึม ผู้วิจัยได้ใช้ประโยชน์จากการประมวลผลแบบขนานเพื่อจัดการทรัพยากรต่างๆ ในระบบเพื่อดึงประสิทธิภาพของระบบออกมาใช้ให้ได้มากที่สุด ผู้วิจัยได้เสนอวิธีการนำ process หรือ หน่วยการทำงานย่อย เป็นตัวแทนของอนุภาคแต่ละอนุภาคในการประมวลผล และใช้การ synchronization เพื่อจัดการการแลกเปลี่ยนข้อมูลและทรัพยากรระหว่างหน่วยประมวลผลย่อย เพื่อที่จะประเมินประสิทธิภาพที่เพิ่มปัญหา travelling Salesman Problem (TSP) ถูกเลือกมาเป็นโจทย์ปัญหาใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

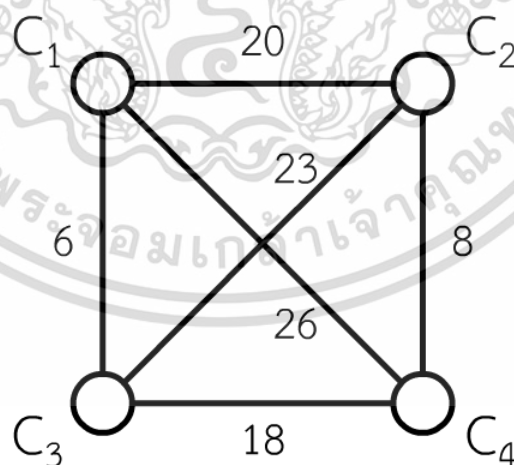
การพิสูจน์ทฤษฎี ว่าสามารถลดเวลาในการประมวลผลจากอัลกอริทึมแบบต้นฉบับได้หรือไม่ ผลของการทดลองจะแสดงให้เห็นถึงประสิทธิภาพในด้านต่างๆ ที่เพิ่มขึ้น

4.2.1 ปัญหาการเดินทางของนักขาย (Traveling salesman problem: TSP)

TSP เป็นที่รู้ในนามของ hard-NP problem จึงถูกนำมาใช้ในการประเมินประสิทธิภาพของอัลกอริทึม โดยกำหนดให้ในระบบมีเมืองหลาย ๆ เมืองที่เชื่อมต่อกันทุกเมือง โดยที่แต่ละเมืองจะมีเส้นทางที่เชื่อมต่อสู่กันและกันเสมอ

เป้าหมายของปัญหานี้คือหาเส้นทางที่จะสามารถผ่านเมืองทุกเมืองที่ไม่ซ้ำกันและกลับมายังเมืองแรกโดยที่ได้ระยะทางรวมที่สั้นที่สุดและไม่สามารถผ่านเมืองที่เคยผ่านมาแล้วได้ ในรูปภาพที่ 4.1 คือตัวอย่างของชุดข้อมูลทดสอบของปัญหานี้ ในตัวอย่างมีเมืองอยู่ 4 เมืองด้วยกันคือ C1 ถึง C4 ระยะทางระหว่างเมืองจะเป็นไปตามที่ปรากฏในรูปภาพ คำตอบของคำถามชุดนี้คือต้องเดินทางไปยัง C1 C3 C4 และ C2 และกลับมาที่เมือง C1 ตามลำดับ จะได้ระยะทางรวมคือ 52

การที่จะแก้ไขปัญหานี้ด้วยวิธีพื้นฐานหรือ วิธีทางฮิวริสติก (Heuristic method) จะต้องหากรณีที่เป็นได้ทั้งหมดที่จะเกิดขึ้นและนำระยะทางของเส้นทางที่เป็นไปได้ทั้งหมดมาเปรียบเทียบ ระยะทางรวมจะทำให้ระยะเวลาของการคำนวณชุดข้อมูลของระบบอยู่ในรูปแบบของ แฟกทอเรียล (Factorial) นั่นหมายความว่าทำให้ Big-O อยู่ในรูปแบบของ n ยกกำลัง n เนื่องจาก n คือจำนวนของเมืองทั้งหมด การคำนวณแบบพื้นฐานจึงเป็นไปได้เลยที่จะนำมาใช้ในการแก้ปัญหานี้ทั้งในแง่ของเวลาในการประมวลผลและหน่วยความจำ หนึ่งในวิธีที่สามารถแก้ไขปัญหานี้ก็คือ optimization



รูปที่ 4.1 ตัวอย่างปัญหาการเดินทางของนักขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การแก้ไขปัญหาคับด้วยวิธี 2-opt swap

2-opt swap คืออัลกอริทึมการค้นหาที่ถูกคิดค้นโดย Croes ในปี ค.ศ. 1958 เพื่อที่จะแก้ไขปัญหาคับการเดินทางของนักขาย หลักการของวิธีนี้คือการสุ่มเส้นทางย่อยมา 2 เส้นทาง และลองสลับกัน จากนั้นลองคำนวณระยะทางรวมใหม่ทั้งหมด หากได้คำตอบที่ดีขึ้นจะยึดเอาเส้นทางใหม่มาใช้ต่อไป หากไม่เป็นเช่นนั้นจะย้อนกลับไปใช้เส้นทางเดิมระบบจะทำการสุ่มแบบนี้ไปเรื่อย ๆ จนกว่าจะถูกสั่งให้หยุด

อัลกอริทึมนี้ไม่รับประกันว่าจะระบบจะพบกับคำตอบที่ดีที่สุดเนื่องจากเป็นเพียงการสุ่มไปเรื่อย ๆ เท่านั้น ระบบอาจจะพบเพียงแค่ส่วนหนึ่งของคำตอบที่ดีที่สุด (Sub optimum) แต่อย่างไรก็ตามวิธีนี้สามารถหาคำตอบใกล้เคียงกับคำตอบที่ดีที่สุดได้ภายในเวลาอันสั้น

4.2.3 การประยุกต์ใช้อัลกอริทึม 2-opt ด้วย PSO

ในการประยุกต์ใช้อัลกอริทึม PSO เพื่อให้ใช้กับ 2-opt swap ได้นั้น อนุภาคการคำนวณแต่ละอนุภาคถูกออกแบบให้สามารถเก็บเส้นทางการเดินทางเพื่อเป็นคำตอบหรือค่าสูงสุดสัมพัทธ์ v_i หรือความเร็วของอนุภาคถูกออกแบบให้เป็นข้อจำกัดของการยอมรับของคำตอบส่วน ค่าสูงสุดของระบบ หรือ Gbest ถูกออกแบบให้เก็บเส้นทางที่สั้นที่สุดหรือคำตอบของระบบนั่นเอง การคำนวณในแต่ละอนุภาคการคำนวณจะเป็นไปตามรหัสเทียมดังรูปที่ 4.2

```

Initialize all particles ; (1)
Loop until timeout ; (2)
  For i to number of particles ; (3)
    Random 2 edge in  $X_i$  as  $C_i$  ; (4)
    If(result ( $C_i$ ) satisfied  $V_i$ ) ; (5)
      Update  $X_i$  ; (6)
      Update pbest $_i$  gbest ; (7)
    Update  $V_{all}$  ; (8)

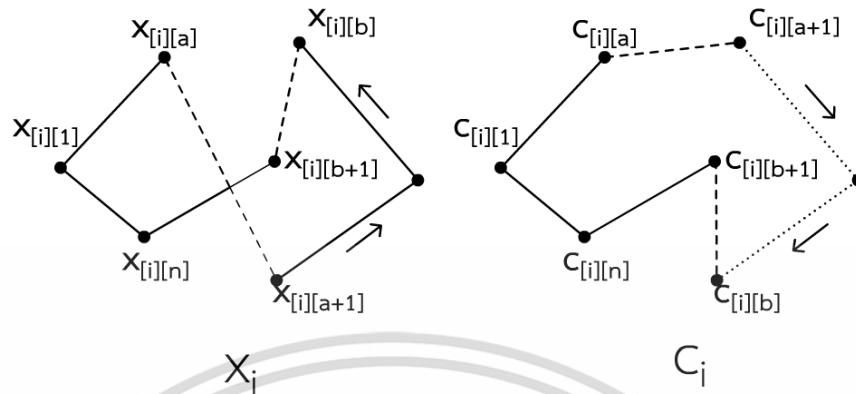
```

รูปที่ 4.2 รหัสเทียมของการประยุกต์ใช้อัลกอริทึม 2-opt ด้วย PSO

รหัสนี้บรรยายถึงการทำงานของอัลกอริทึม 2-opt ที่ได้กล่าวไปข้างต้น c_i คือ คำตอบที่เป็นตัวเลือก (candidate solution) ที่สร้างมาจากอัลกอริทึม 2-opt เพื่อที่จะสร้างคำตอบที่เป็นตัวเลือกขึ้นมา เส้นทางย่อย 2 เส้นทางจะถูกสุ่มเลือกออกมา และสลับกันจากนั้นจะเข้ากระบวนการจัดการรูปแบบ (Reconstruct) เพื่อให้ทิศทางการเดินทางของเส้นทางนั้นถูกต้อง

รูปที่ 4.3 แสดงถึงการสุ่มสร้างคำตอบใหม่ขึ้นมา (c_i) ของอนุภาคการคำนวณหมายเลขที่ i เปรียบเทียบกับคำตอบเดิม (x_i) โดยที่ n คือจำนวนของเมืองทั้งหมดในปัญหา a กับ b คือหลายเลข

ของเส้นทางย่อยที่ถูกสุ่มขึ้นมาเพื่อทำงานตามวิธี 2-opt ซึ่งตัวแปรที่สุ่มขึ้นมาจะต้องเป็นไปตามเงื่อนไข $0 < a+1 < b+1 < n$



รูปที่ 4.3 การสุ่มสร้างคำตอบใหม่ของอัลกอริทึม 2-opt ด้วย PSO

เมื่อสุ่มตัวแปร a และ b ขึ้นมา เส้นทางย่อยที่จะถูกสับเปลี่ยนกันคือเส้นทางระหว่างเมืองที่ $x[i][a]$ ไปยัง $x[i][a+1]$ จะถูกสับเปลี่ยนกับเส้นทางระหว่างเมืองที่ $x[i][b]$ ไปยัง $x[i][b+1]$ (ดังที่แสดงในรูปที่ 4.3) หลังจากสับเส้นทางทั้งสองแล้วจะต้องสลับทิศทางการเดินทางจาก $x[i][b]$ ไปยัง $x[i][a+1]$ ด้วยเพื่อให้การคำนวณมีทิศทางที่ต้องการ

คำตอบใหม่ที่สุ่มขึ้นมา (c_i) จะถูกนำมาคำนวณระยะทางรวมใหม่และนำไปเปรียบเทียบกับ G_{best} หรือค่าสูงสุดสัมพัทธ์เพื่อตัดสินใจว่าจะเปลี่ยนแปลงคำตอบปัจจุบันหรือไม่ถ้าเกิดการเปลี่ยนแปลงที่ดีขึ้น G_{best} หรือค่าสูงสุดสัมพัทธ์จะถูกเปลี่ยนแปลงไปด้วย หลังจากการคำนวณเสร็จในแต่ละรอบคำตอบของแต่ละอนุภาคจะนำมาเปรียบเทียบกับค่าสูงสุดของระบบหรือ G_{best} เพื่อเปลี่ยนแปลงคำตอบที่ดีที่สุด

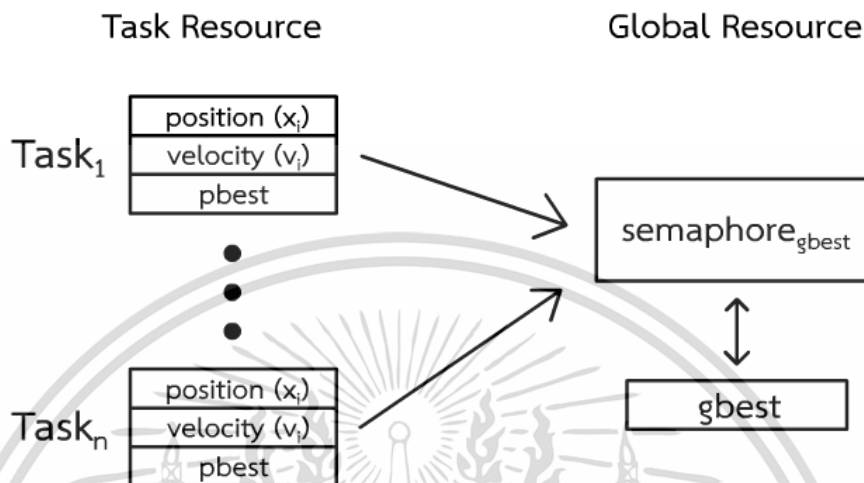
4.2.4 การคำนวณ PSO แบบ Multithread

การคำนวณแบบขนานสามารถทำได้โดยการใช้เทคนิค multithread เพื่อลดเวลาการประมวลผล ในอัลกอริทึม PSO แต่ละอนุภาคการคำนวณจะทำงานอิสระจากกัน ยกเว้นค่าสูงสุดของระบบหรือ G_{best} จะเป็นเพียงทรัพยากรเดียวที่ใช้ร่วมกัน ดังนั้นเราสามารถใช้ประโยชน์จากการแยกกันอย่างอิสระของอนุภาคการคำนวณเพื่อให้แต่ละอนุภาคการคำนวณทำงานแบบขนานกัน การประยุกต์เช่นนี้จะทำให้ระบบพบกับคำตอบที่ดีในเวลาเร็วขึ้น

ระบบที่ผู้วิจัยเสนอขึ้นมานี้จะกำหนดให้หนึ่งอนุภาคการทำงานคือ task หรือหนึ่งหน่วยทำงานของอนุภาคการคำนวณในแต่ละ task จะประกอบไปด้วย ตำแหน่ง (เส้นทางของการเดินทางที่เป็นคำตอบ) ความเร็ว และจะมีการทำงานจากหลักการของอัลกอริทึม 2-opt PSO ในบรรทัดที่ (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงบรรทัดที่ (7) ของรหัสเทียมในรูปที่ 4.2 ระบบจะต้องสร้าง task ขึ้นมาให้มีจำนวนเท่ากับจำนวนของอนุภาคการคำนวณ และมีการใช้ทรัพยากรร่วมกันในส่วนของค่าสูงสุดของระบบ หรือ Gbest หรือก็คือคำตอบของระบบนั่นเอง

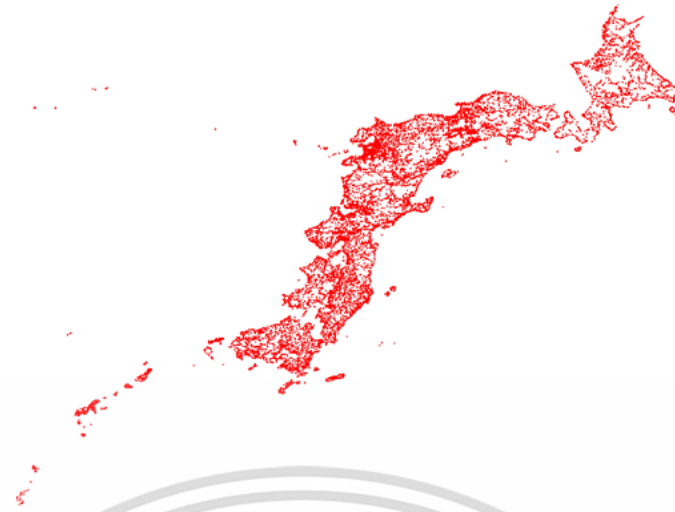


รูปที่ 4.4 ตัวอย่างปัญหาการเดินทางของนกชาย

รูปที่ 4.4 แสดงให้เห็นถึงทรัพยากรที่มีอยู่ในแต่ละ task อันประกอบด้วย x_i v_i และ P_{best} รวมไปถึง G_{best} ที่เป็นทรัพยากรที่ใช้ร่วมกันด้วย เมื่อระบบเริ่มต้นการทำงานระบบจะทำการกำหนดค่าเริ่มต้นของทรัพยากรทั้งหมดในระบบ (Initialization) และเริ่มการทำงานของทุก task พร้อมกันจะพบว่าระบบทำงานได้เร็วขึ้น แต่อย่างไรก็ตามระบบจะไม่สามารถทำงานเร็วขึ้นได้เมื่อจำนวนของอนุภาคการคำนวณมีมากกว่าจำนวนของหน่วยประมวลผลในระบบ (Logical processors)

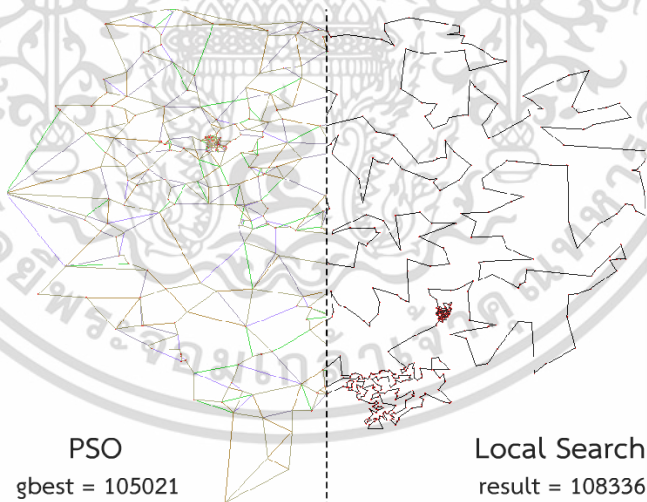
4.2.5 การทดลอง

ชุดของปัญหาการเดินทางของนกชายที่นำมาใช้นั้นอยู่ในรูปแบบของ TSPLIB จะเป็นจุดบนระนาบ 2 มิติที่เป็นตัวแทนของเมืองในประเทศต่างๆ ซึ่งสามารถคำนวณหาระยะห่างระหว่างเมืองได้ทุกเมือง [17]



รูปที่ 4.5 ตัวอย่างของชุดข้อมูลปัญหา (ja9847)

รูปที่ 4.5 แสดงให้เห็นตัวอย่างของชุดข้อมูลโจทย์ปัญหา ในที่นี้ได้ใช้ชุดข้อมูลที่มีชื่อว่า ja9847 ประกอบไปด้วยเมืองทุกเมืองของประเทศญี่ปุ่นจำนวน 9847 เมือง จุดสีแดงในรูปที่ 4.5 คือตัวแทนของเมืองแต่ละเมือง และทุกเมืองสามารถหาระยะห่างระหว่างกันได้โดยการใช้หลักการระยะห่างระหว่างจุดได้

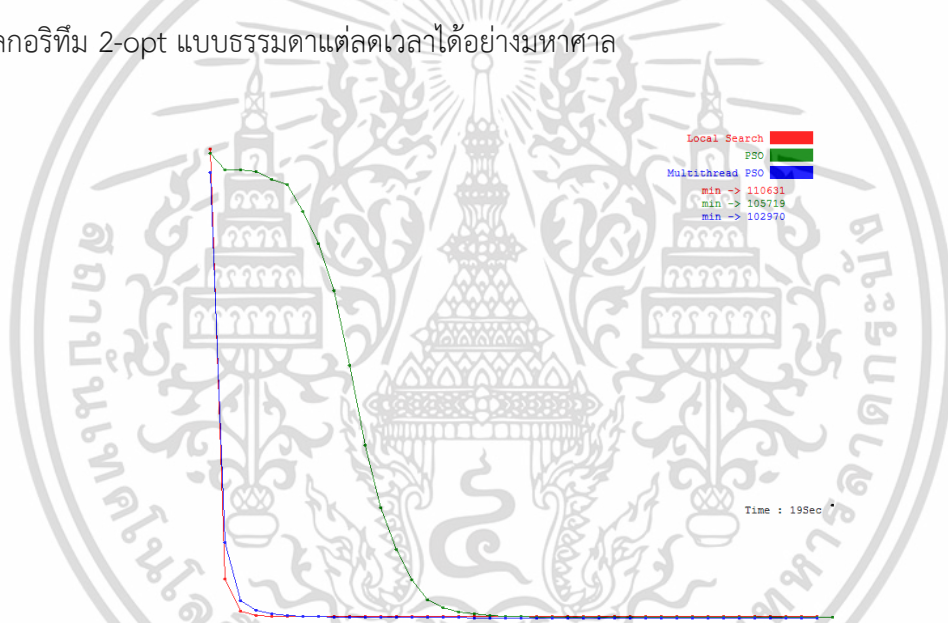


รูปที่ 4.6 แสดงคำตอบของระบบของชุดคำถาม (zi929) ของอัลกอริทึม PSO (ด้านซ้าย) และ 2-opt (ด้านขวา)

รูปที่ 4.6 แสดงผลลัพธ์ของโปรแกรมที่ผู้วิจัยได้พัฒนาขึ้นโดยกำหนดเวลาให้ทำงาน 20 วินาที โดยใช้ชุดข้อมูลทดสอบ zi929 หรือข้อมูลของเมืองในประเทศซิมบับเว ที่มีเมืองจำนวน 929 เมืองด้วยกัน โดยที่คำตอบที่ดีที่สุดของคำถามชุดนี้คือ 95345 รูปทางขวาแสดงผลลัพธ์จากอัลกอริทึม 2-เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

opt แบบธรรมดาจึงมีเพียงแค่ 1 เส้นทางในเวลาหนึ่งๆ ผลคือได้ระยะทางรวมเป็น 108336 (คิดเป็น 88.00% ของคำตอบที่ดีที่สุด) ภาพทางซ้ายคือผลลัพธ์จากการประยุกต์อัลกอริทึม PSO เข้าไปช่วยทำงานด้วย สีแต่ละสีเป็นตัวแทนของอนุภาคการคำนวณแต่ละตัว ในตัวอย่างข้างต้นได้ใช้อนุภาคการคำนวณ 5 ตัว ผลคือได้ระยะทางรวมเป็น 105021 (คิดเป็น 90.79% ของคำตอบที่ดีที่สุด) ซึ่งจะเห็นว่าจะได้คำตอบที่ดีที่สุดว่า อัลกอริทึม 2-opt แบบ ธรรมดา

รูปที่ 4.7 แสดงให้เห็นถึงเวลาในการประมวลผลที่ลดลงของการหาคำตอบ กราฟสีแดงคืออัลกอริทึม 2-opt แบบธรรมดา ซึ่งจะใช้เวลาไม่น้อยที่สุดก่อนที่ระบบจะหยุดนิ่ง สีเขียวคือการใช้ PSO เข้าไปช่วยในการทำงานทำให้ได้คำตอบที่ดีขึ้นแต่ต้องแลกมาด้วยเวลาที่นานขึ้น (ในตัวอย่างนี้ใช้อนุภาคการคำนวณ 5 อนุภาคด้วยกัน) ซึ่งเวลาที่เพิ่มขึ้นมานั้นจะมีค่าประมาณจำนวนของอนุภาคการคำนวณคูณด้วยเวลาของการคำนวณแบบธรรมดา กราฟสีน้ำเงินคือวิธีที่ผู้วิจัยได้พัฒนาให้ระบบสามารถคำนวณได้เร็วยิ่งขึ้นจะเห็นได้ว่าผลลัพธ์จะมีค่าเทียบเท่ากับการใช้ PSO ซึ่งดีกว่าการใช้อัลกอริทึม 2-opt แบบธรรมดาแต่ลดเวลาได้อย่างมหาศาล



รูปที่ 4.7 กราฟแสดงเวลาและผลลัพธ์ที่ได้จากการคำนวณ 2-opt

4.3 การประมวลผลค่าที่เหมาะสมที่สุดแบบไม่สุ่มเพื่อคำตอบที่คงที่ (Hard Deterministic Particle Swarm Optimization for Certain Result Solution, HD-PSO)

การประมวลผลด้วยการสุ่ม (Stochastic method) อาจจะทำให้ผลลัพธ์ที่ดีมากแต่ปัญหาสำคัญปัญหาหนึ่งของการประมวลผลแบบนี้ก็คือผลลัพธ์จะเกิดแบบสุ่ม เนื่องจากการประมวลผลทั้งหมดเกิดขึ้นจากการสุ่ม หมายความว่าคำตอบที่ออกมา นั้นไม่สามารถที่จะคาดเดาได้ (Unpredictable result) การปรับตัวแปรหรือการพัฒนาวิธีการต่าง ๆ เป็นเพียงการปรับแนวโน้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการสุ่มให้เป็นไปตามที่ต้องการ [18] แต่ไม่ได้เกี่ยวข้องโดยตรงกับการคำนวณ ผลลัพธ์ของการประมวลผลจะแตกต่างกันไปทุกครั้งที่ทำการเริ่มโปรแกรม ถึงแม้ว่าส่วนใหญ่จะให้ผลลัพธ์ที่ดีและยอมรับได้แต่ก็มีโอกาสเล็กน้อยที่จะเกิดผลลัพธ์ที่แย่และไม่สามารถยอมรับได้ เหตุการณ์นี้สามารถเกิดขึ้นได้ตลอดเวลาที่ระบบทำงาน เป็นเหตุผลให้การวิเคราะห์ประสิทธิภาพของวิธีการต้องทำการเริ่มโปรแกรมหลายครั้งและบันทึกผลเพื่อใช้ค่าทางสถิติ โดยไม่สามารถใช้ผลลัพธ์จากการทำงานเพียงครั้งเดียวได้

ผลเสียอีกประการของการประมวลผลโดยการสุ่มคือ ความยากในการวิเคราะห์ การวิเคราะห์ผลลัพธ์ที่เกิดจากการสุ่มนั้นทำได้โดยวิธีการทางสถิติเท่านั้น อีกทั้งการสุ่มในระบบประมวลผลส่วนใหญ่เกิดขึ้นจากการสุ่มแบบเทียม (Pseudo-random) ซึ่งทำได้โดยการนำลำดับที่มีอยู่แล้วมาแสดงผล จึงทำให้การวิเคราะห์เป็นไปได้ยากขึ้น

ผู้วิจัยจึงได้พัฒนาอัลกอริทึมการหาค่าเหมาะสมที่สุดที่ไม่ใช่การสุ่มเพื่อลดปัญหาดังกล่าว ในงานวิจัยนี้ผู้วิจัยได้เลือกการหาค่าเหมาะสมที่สุดที่เป็นที่นิยมในปัจจุบัน คือการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคมาทำการเปลี่ยนแปลงวิธีการคำนวณจากเดิมใช้การสุ่มเป็นหลักให้เป็นการคำนวณโดยให้ผลลัพธ์ที่แน่นอน สุดท้ายจะเปรียบเทียบผลลัพธ์กับอัลกอริทึมต้นแบบและอภิปรายผลที่เกิดขึ้นจากการไม่สุ่ม

4.3.1 การหาค่าเหมาะสมแบบอนุภาคที่สุดโดยไม่สุ่ม

ผู้วิจัยมุ่งเน้นไปที่การปรับเปลี่ยนการเคลื่อนที่ของอนุภาค จากการที่ใช้การสุ่มเป็นหลักในการคำนวณให้เป็นการคำนวณแบบมีผลลัพธ์ที่แน่นอน เพื่อที่จะให้เป็นไปตามจุดหมายที่ตั้งไว้ ผู้วิจัยได้คิดค้นวิธีการขึ้นมา 3 ขั้นตอนด้วยกัน คือ การแบ่งขอบ (Bound slicing) การใช้หน้าต่างเคลื่อนที่ (Windows moving) และการผ่าเหล่าแบบตามลำดับ (Sorting mutation)

4.3.1.1 การแบ่งขอบ (Bound slicing)

การแบ่งขอบหมายถึงอัลกอริทึมที่สร้างขึ้นมาเพื่อกำหนดจุดเริ่มต้นของประชากรหรืออนุภาคทั้งหมดในระบบโดยปกติแล้วอัลกอริทึมการหาค่าเหมาะสมที่สุดที่มีหลายหน่วยย่อยในการทำงานจะกำหนดจุดเริ่มต้นของการทำงานแบบสุ่ม [19] อนุภาคในระบบการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค อนุภาคแต่ละตัวจะมีตำแหน่งเริ่มต้นด้วยการสุ่ม แต่ในกรณีนี้เราต้องการทำให้การเริ่มต้นมีจุดที่แน่นอนในทุก ๆ การคำนวณ วิธีนี้จะเริ่มจากรูปแบบมาตรฐานของปัญหาการหาค่าเหมาะสมที่สุดซึ่งเขียนให้อยู่ในรูปของสมการที่ (4.1)

$$\text{minimize } f(x) : lb_i < x_i < ub_i \quad (4.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ lb_i และ ub_i คือขอบด้านล่างและขอบด้านบนสุดของแต่ละมิติในปัญหา การแบ่งขอบ จะเริ่มจากการสมมติปัญหาให้เป็นรูปทรงที่มีขอบบนและล่างตามขอบของปัญหา การแบ่งขอบจะทำให้เป็นรอบ ๆ โดยแต่ละรอบจะแบ่งครึ่งโดยยึดจากจุดศูนย์กลางของรูปทรงเป็นหลักในแต่ละรอบ ในรอบแรกสุดจะถือว่าให้มีรูปทรงหนึ่งอันและจะแบ่งขอบที่มีความกว้างมากที่สุดให้กลายเป็นรูปทรง 2 รูปทรง ตามรหัสเทียมในรูปที่ 4.8

```

1 procedure Bound Slicing (pop[][], bound[], NP)
2   Let box contain ub[] and lb[]
3   Let inputQ and outputQ be a queue of box
4   Init pop[0] with bound and insert into inputQ
5   Set Remainslice = NP
6   While Remainslice > 0
10  While inputQ is not empty and Remainslice > 0
11    Set Upiece and Lpiece = inputQ[0]
12    Set center = (Upiece.ub[] + Upiece.lb[]) / 2
13    Set Upiece.lb[widest] = center
14    Set Lpiece.ub[widest] = center
15    Enqueue first inputQ
16    Insert Upiece and Lpiece into outputQ
17    Remainslice = Remainslice - 1
18    Sort outputQ and insert into inputQ
19  Concatenate outputQ with inputQ
20  Return outputQ

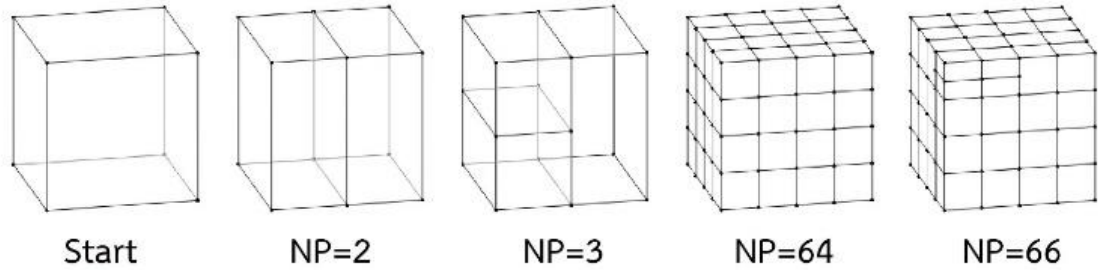
```

รูปที่ 4.8 รหัสเทียมของการแบ่งขอบ

จากรหัสเทียม แสดงให้เห็นถึงวิธีการที่ใช้ในการประมวลผลของการแบ่งขอบ หากสมมติให้ปัญหามีมิติคือ 3 เราจะสามารถมองขอบทั้งหมดเป็นรูปทรง 3 มิติได้ ซึ่งเริ่มต้นจะมีรูปทรงลูกบาศก์ 1 รูปทรง และจะเริ่มการแบ่งให้เป็น 2 รูปทรง ในรอบแรกโดยใช้จุดศูนย์กลาง รูปที่ 4.9 จะแสดงให้เห็นถึงการแบ่งเป็นภาพสามมิติ

รูปที่ 4.9 แสดงให้เห็นตัวอย่างของการแบ่งขอบตามจำนวนของประชากรหรืออนุภาค ใด ๆ หากจำนวนประชากรมีค่าเท่ากับ $(dim^2)^n$ นั้น รูปทรงที่แบ่งออกมาจะเป็นรูปที่เท่ากัน (ในปัญหาที่มีมิติเท่ากับ 3 จะมีค่าเท่ากับ 8 64 512 ...) หากไม่เท่ากับจำนวนดังกล่าวจะเกิดรูปทรงที่ไม่เท่ากันเกิดขึ้น เช่นในรูปที่ประชากรเป็น 2 และ 66 จะเห็นได้ว่าเกิดรูปทรงสี่เหลี่ยมเล็กที่มีปริมาตรไม่เท่ากับลูกอื่นเกิดขึ้น

การแบ่งขอบนี้จะทำให้ได้รูปทรงที่มีจำนวนเท่ากับจำนวนประชากรเราจะใช้จุดศูนย์กลางของขอบย่อยแต่ละขอบเพื่อเป็นตำแหน่งเริ่มต้นให้กับอนุภาคในอัลกอริทึมต่อไป วิธีนี้จะทำให้อนุภาคมีการกระจายตัวอย่างทั่วถึงในเวลาที่ยกกัดและจะให้ผลลัพธ์ที่แน่นอนทุกครั้งที่คำนวณ

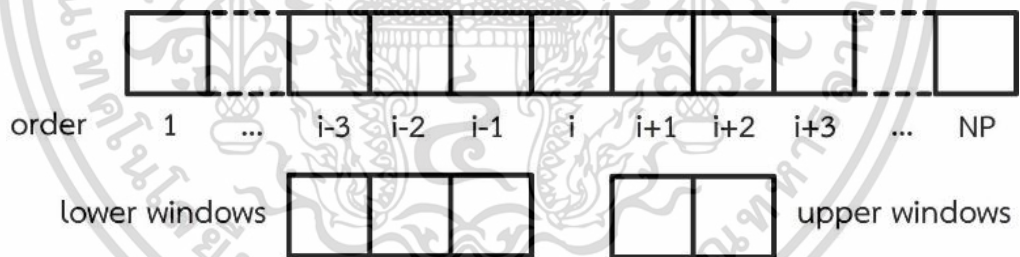


รูปที่ 4.9 ตัวอย่างการแบ่งขอบในปัญหา 3 มิติ

4.3.1.2 การใช้หน้าต่างเคลื่อนที่ (Windows moving)

อัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบทั่วไปนั้น การเคลื่อนที่ของอนุภาคแต่ละตัวจะถูกกำหนดด้วย 2 สิ่งคือ ตัวแปรการสุ่มแบบอิสระและตัวแปรการสุ่มที่แนวทางเป็นไปตาม G_{best} ตามสมการที่ 2 เพื่อที่จะกำจัดตัวแปรที่เกิดจากการสุ่มไป การใช้หน้าต่างเคลื่อนที่จึงถูกนำมาใช้เพื่อการได้มาซึ่งคำตอบที่แน่นอน

อนุภาคจะคำนวณตำแหน่งใหม่โดยการใช้ ค่าเฉลี่ยของอนุภาคอื่น ๆ โดยรอบตัวเองโดยจะคำนวณจาก LF (lower factor) อันดับแรกของอนุภาคที่ดีกว่าตนเองและ UF (upper factor) อันดับแรกที่แย่กว่าตนเอง LF จะควบคุมการหาลดลงที่เพิ่มขึ้นในขณะที่ UF จะทำให้ระบบหลุดออกจากจุดที่ต่ำที่สุดเทียมหรือ จุดต่ำสุดสัมพัทธ์ตัวแปรทั้ง 2 ตัวนี้สามารถปรับเปลี่ยนให้เข้ากับปัญหาได้



รูปที่ 4.10 แสดงการทำงานของหน้าต่างเคลื่อนที่

รูปที่ 4.10 แสดงให้เห็นถึงตัวอย่างการคำนวณการใช้หน้าต่างเคลื่อนที่เมื่อให้ประชากรทั้งหมดมี 3 หน่วย กำหนดให้ LF มีค่า 0.3 และ กำหนดให้ UF มีค่า 0.2 จะได้ว่า การคำนวณตำแหน่งใหม่จะใช้ค่าเฉลี่ยของอนุภาคที่ดีกว่า 3 ตัวแรก และอนุภาคที่แย่กว่า 2 ตัวหลังในการคำนวณ ซึ่งหลักการคำนวณจะเป็นไปตามสมการที่ (4.2) และสมการที่ (4.3)

$$x_i^{cd} = \left(x_i^t \times (1 - v_i^t) \right) + \left(windows_{avg} \times v_i^t \right) \quad (4.2)$$

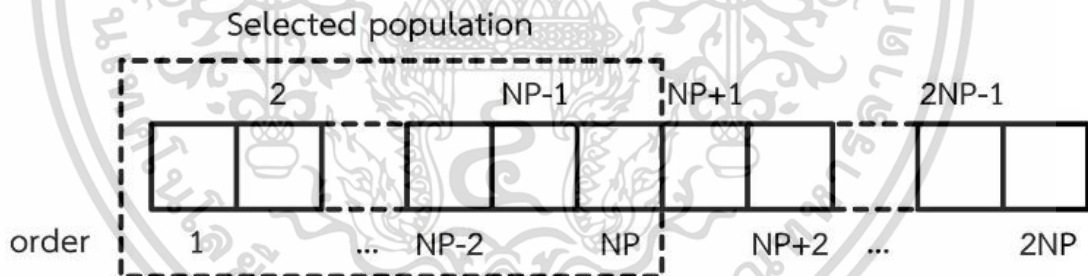
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v_i^{t+1} \begin{cases} \text{better: } v_i^t \times 0.99 \\ \text{worse: } v_i^t \times 1.01 \end{cases} \quad (4.3)$$

สมการที่ (4.2) แสดงให้เห็นการคำนวณตำแหน่งใหม่หรือที่เรียกว่า candidate solution (x_i^{cd}) ซึ่งจะมีการเปลี่ยนแปลงความเร็วไปตามแนวโน้มที่ดีขึ้นและแย่ลงของการหาคำตอบไปเรื่อย ๆ ตามสมการที่ (4.3) แต่เนื่องจากการคำนวณแบบนี้ให้คำตอบที่แน่นอนทุกครั้งจึงเกิดปัญหาที่ทำให้ระบบไปหยุดที่จุด ๆ หนึ่ง ซึ่งไม่ว่าจะคำนวณอย่างไรก็จะได้ผลลัพธ์เท่าเดิม ทำให้ผลลัพธ์ที่ออกมา นั้นไม่มีทางดีขึ้นได้ จึงต้องใช้เทคนิคการผ่าเหล่าแบบตามลำดับ (Sorting mutation) เพื่อให้ระบบสามารถไปยังจุดที่ดีขึ้นได้

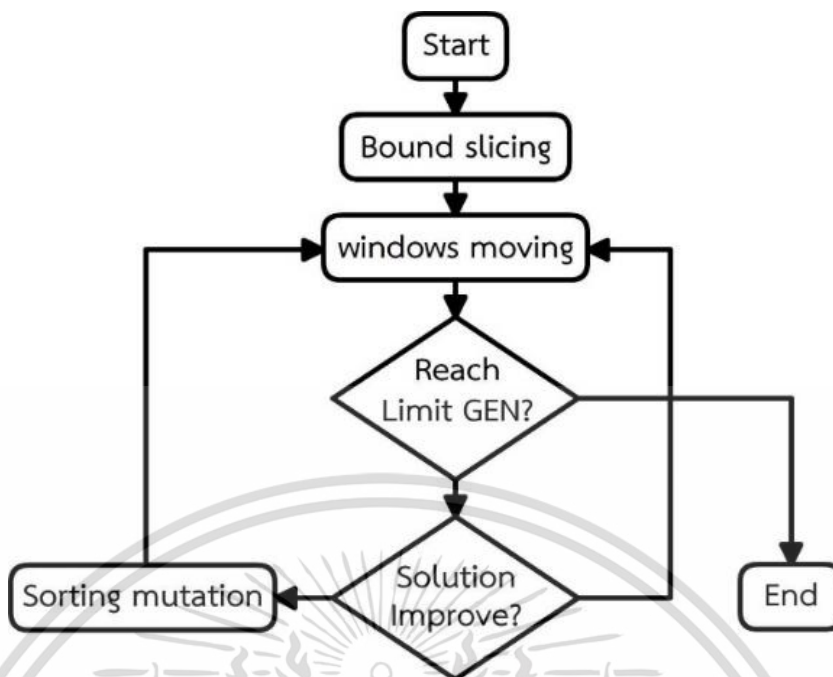
4.3.1.3 การผ่าเหล่าแบบตามลำดับ (Sorting mutation)

เมื่อระบบไม่สามารถหาคำตอบที่ดีขึ้นได้ด้วยวิธีการใช้หน้าตาที่เปลี่ยน การผ่าเหล่าแบบตามลำดับจะถูกเรียกใช้งานโดยจะทำการนำประชากรทั้งหมดมารวมกับผลลัพธ์ในรอบใหม่ (Candidate solution) และทำการเรียงลำดับจากผลลัพธ์ที่ดีที่สุดไปยังผลลัพธ์ที่แย่ที่สุด จากนั้นจะเลือกเฉพาะประชากรครั้งแรกที่มีผลลัพธ์ที่ดีมาตั้งต้นเป็นประชากรชุดใหม่ในการคำนวณครั้งต่อไป ตามรูปที่ 4.11



รูปที่ 4.11 การเลือกประชากรในรอบต่อไป

ระบบจะทำงานไปเรื่อย ๆ ตามจำนวนรอบที่ตั้งค่าไว้และจะสลับการใช้ การใช้หน้าตาที่เปลี่ยนที่และการผ่าเหล่าแบบตามลำดับ ตามเงื่อนไขดังผังงานในรูปที่ 4.12



รูปที่ 4.12 ผังงานแสดงการทำงานของทั้งระบบ

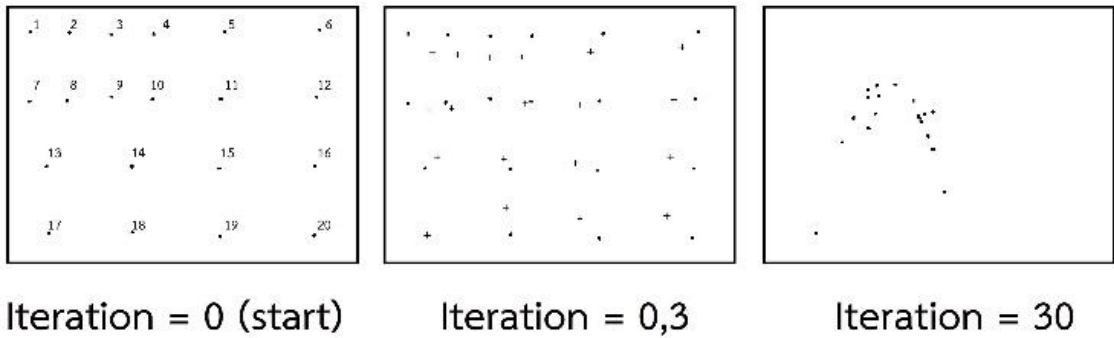
4.3.2 การทดลองและผลลัพธ์

การประยุกต์และการทดลองทั้งหมดของวิธีการที่พัฒนาขึ้นมาได้ทดลองทำบน ระบบที่มีชื่อว่า Pagmo [20] Pagmo เป็นชื่อของชุดซอฟต์แวร์ (Platform) ที่พัฒนาขึ้นมาเพื่อการหาค่าเหมาะสมที่สุดโดยจะบรรจุทั้ง อัลกอริทึม โจทย์ปัญหา และเครื่องมือสำหรับวิเคราะห์ไว้ด้วยกัน โดยใช้ภาษา C++

การทดลองจะแบ่งออกเป็น 2 ส่วนหลักคือ อย่างแรกจะทดลองในเรื่องของผลลัพธ์ที่แน่นอนว่า วิธีการที่พัฒนาขึ้นมานั้นให้ผลลัพธ์ที่แน่นอนจริงหรือไม่และมีผลลัพธ์ในรูปแบบใด ส่วนที่สองจะเป็นเรื่องของประสิทธิภาพเมื่อเทียบกับอัลกอริทึมอื่น

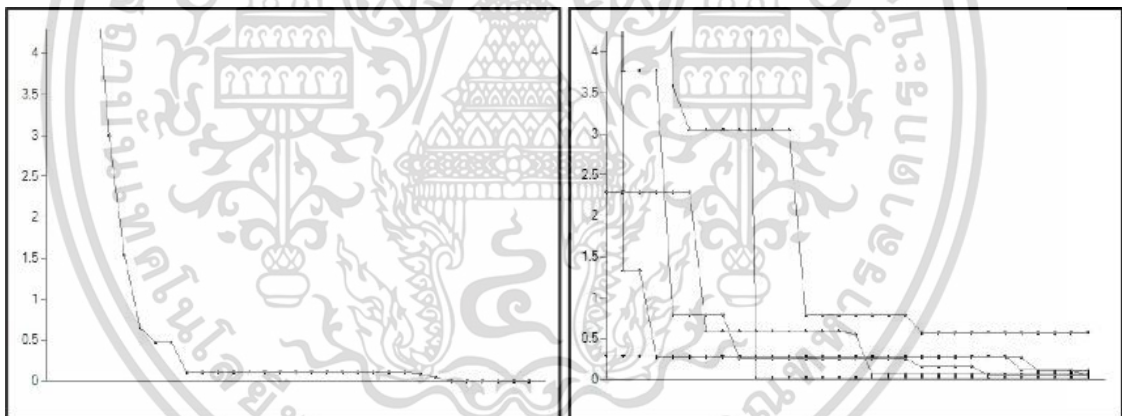
4.3.2.1 ลักษณะเฉพาะของ HD-PSO

ในการทดลองแรกนั้นเพื่อความเข้าใจง่ายขึ้นจะใช้โจทย์ปัญหาที่มีความซับซ้อนต่ำในการทดลองก่อน ในที่นี้จะใช้ฟังก์ชัน Rosenbrock 2 มิติ ในการทดลอง ตำแหน่งของอนุภาคและผลลัพธ์จะถูกบันทึกไว้เพื่อการวิเคราะห์



รูปที่ 4.13 ตำแหน่งของอนุภาคเมื่อเริ่มทำงาน

รูปที่ 4.13 แสดงถึงตำแหน่งของอนุภาคทั้งหมดระหว่างที่ระบบกำลังประมวลผล ณ รอบต่าง ๆ เมื่อตั้งค่าให้จำนวนประชากรทั้งหมดมี 20 หน่วย ภาพซ้ายสุดแสดงตำแหน่งเริ่มต้นของอนุภาคทุกตัวซึ่งเกิดขึ้นจากขั้นตอนของการแบ่งขอบ จากที่ได้กล่าวไว้ข้างต้นเกี่ยวกับขนาดที่ไม่เท่ากันของรูปทรงที่ถูกแบ่งจะเห็นได้ว่าอนุภาคหมายเลข 5 6 10 11 12 13 14 15 16 17 18 19 และ 20 จะมีระยะห่างที่เท่ากันส่วนอนุภาค 1 2 3 4 7 8 9 10 จะมีระยะห่างที่ไม่เท่ากัน



รูปที่ 4.14 กราฟเปรียบเทียบค่าคำตอบระหว่าง HD-PSO และ PSO แบบธรรมดา

รูปที่ 4.14 แสดงกราฟของผลลัพธ์ที่เกิดขึ้นจากการประมวลผลทั้งหมด 10 ครั้ง โดยรูปด้านซ้ายคืออัลกอริทึมที่ผู้วิจัยพัฒนาขึ้นมาจะเห็นได้ว่าผลลัพธ์ที่เกิดขึ้นมานั้นมีค่าที่แน่นอน กล่าวคือมีผลลัพธ์ของการประมวลผลของทุกอนุภาคเท่าเดิมทุกครั้งในทุกรอบของการประมวลผล ส่วนรูปด้านขวาคือ ภาพที่เกิดจากอัลกอริทึมการหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคทั่วไปซึ่งผลลัพธ์ที่เกิดขึ้นจะต่างกันไปทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2.2 ประสิทธิภาพของอัลกอริทึม HD-PSO

ในการวัดประสิทธิภาพของวิธีที่ได้คิดค้นขึ้นมานั้น จะใช้โจทย์ปัญหาอื่น ๆ ที่มีความซับซ้อนมากขึ้น ดังนั้นผู้วิจัยจึงเลือกฟังก์ชันที่ใช้ทดสอบอย่างแพร่หลายมา 4 ฟังก์ชันด้วยกันได้แก่ Rosenbrock Ackley Lennard Jones และ Griewank ซึ่งในแต่ละฟังก์ชันนั้นจะทำการทดสอบด้วยมิติที่ตั้งค่าไว้ที่ 3 และ 4 ด้วยกัน เพื่อหาผลลัพธ์ที่ออกมาของระบบ และเพื่อเป็นการเปรียบเทียบผลลัพธ์ของวิธีใหม่นี้ ผู้วิจัยได้เลือกอัลกอริทึมที่เป็นที่นิยมในปัจจุบันมาเป็นตัวบ่งชี้ นั่นก็คือ อัลกอริทึม GWO [21] การวิวัฒนาการแบบเปลี่ยนแปลง หรือ Differential Evolution (DE) [22] และ การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบทั่วไป

การทดลองจะทำทั้งหมด 20 ครั้งในแต่ละอัลกอริทึมเพื่อหาค่าที่ดีที่สุด ค่าเฉลี่ย และ ค่าที่แย่ที่สุด เพื่อนำมาเปรียบเทียบกับวิธีที่ผู้วิจัยคิดค้นขึ้น ค่าจำนวนประชากรทั้งหมดถูกตั้งค่าให้เท่ากันที่ 1000 ประชากร จำนวนรอบของการประมวลผลถูกตั้งไว้ที่ 100 รุ่น ส่วนอัลกอริทึมที่ผู้วิจัยพัฒนาขึ้นมานั้นเนื่องจากให้ผลลัพธ์ที่แน่นอนทุกครั้งจึงทำการบันทึกค่าเพียงครั้งเดียว

ตารางที่ 4.1 ผลลัพธ์ของฟังก์ชันทดสอบต่าง ๆ ใน 3 มิติ

		Rosenbrock	Ackley	Griewank	LJ
HD-PSO		2.700E-05	6.746E-04	1.704E-01	-3.000E+00
PSO	avg	1.238E-03	2.917E-04	7.890E-03	-3.000E+00
	best	5.889E-06	9.115E-05	0.000E+00	-3.000E+00
	worst	5.129E-03	6.099E-04	1.616E-02	-3.000E+00
GWO	avg	2.257E-05	7.994E-16	5.061E-03	-3.000E+00
	best	1.579E-06	4.441E-16	0.000E+00	-3.000E+00
	worst	7.428E-05	3.997E-15	1.233E-02	-3.000E+00
DE	avg	5.099E-05	1.129E-03	2.566E-02	-3.000E+00
	best	3.076E-06	3.807E-05	1.549E-02	-3.000E+00
	worst	1.817E-04	2.018E-03	4.726E-02	-3.000E+00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลลัพธ์ของฟังก์ชันทดสอบต่าง ๆ ใน 3 มิติ

		Rosenbrock	Ackley	Griewank	LJ
HD-PSO		1.584E-01	4.555E-03	2.064E-01	-4.150E+00
PSO	avg	6.915E-03	1.668E-03	7.874E-02	-2.738E+00
	best	3.695E-05	6.746E-04	4.953E-02	-3.440E+00
	worst	2.028E-02	2.585E-03	1.179E-01	-2.148E+00
GWO	avg	3.393E-03	1.688E-15	1.312E-02	-2.874E+00
	best	1.282E-05	4.441E-16	0.000E+00	-4.332E+00
	worst	1.043E-02	3.997E-15	3.222E-02	-2.395E+00
DE	avg	5.841E-02	5.511E-03	8.014E-02	-2.771E+00
	best	8.034E-03	2.116E-03	3.923E-02	-3.713E+00
	worst	1.267E-01	9.486E-03	1.079E-01	-2.040E+00

ตารางที่ 4.1 และตารางที่ 4.2 นั้น คือสถิติที่บันทึกไว้สำหรับการทดลองที่กล่าวมาข้างต้น ค่าที่น้อยกว่าแสดงถึงผลลัพธ์ที่ดีกว่า ผลที่ออกมาที่น้อยนั้นเป็นไปตามที่ผู้วิจัยคาดการณ์ไว้เนื่องจาก อัลกอริทึมที่ได้คิดค้นขึ้นมานั้นสามารถทำงานได้ตามที่ต้องการโดยที่ได้ค่าออกมาใกล้เคียงกับผลเฉลี่ยของอัลกอริทึมที่ใช้ในปัจจุบัน ซึ่งที่สำคัญยังให้ผลลัพธ์ที่มีค่าดีกว่าค่าที่แย่ที่สุดของเกือบทุกการทดลอง แต่ยังไม่สามารถให้ผลลัพธ์ที่ดีไปกว่าค่าที่ดีที่สุดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การหาค่าที่เหมาะสมที่สุดแบบจุดหมุน (Pivot Optimiser)

5.1 บทนำ

ในช่วงหลายปีที่ผ่านมา อัลกอริทึมการหาค่าที่ดีที่สุดได้มีการพัฒนาต่อกันมา วิธีการส่วนใหญ่ได้ต้นแบบมาจากธรรมชาติและทฤษฎีต่าง ๆ เช่น GWO และ PSO เป็นต้น แต่ละวิธีการนั้นได้พัฒนาตามกลไกการทำงานของทฤษฎีต้นแบบและพัฒนาให้เข้ากับการหาค่าที่ดีที่สุด งานวิจัยนี้มีชื่อว่า การหาค่าที่เหมาะสมที่สุดแบบจุดหมุน หรือ Pivot Optimiser คืออัลกอริทึมการหาค่าที่ดีที่สุดแบบการวิวัฒนาการ (Evolution optimisation algorithm) ที่ได้ต้นแบบมาจากทฤษฎีเรขาคณิตแบบหลายมิติ ซึ่งสามารถทำให้เกิดการวิวัฒนาการแบบไม่ซ้ำกันได้ (Unique evolution) เป้าหมายของอัลกอริทึมนี้ คือการทำงานที่ครอบคลุมในทุกสถานการณ์ และการได้มาซึ่งผลลัพธ์ที่มีความเสถียรอยู่ในงานวิจัยนี้มีชื่อเต็มว่า A Novel Evolution Optimization Algorithm Using a Multidimensional Geometric Method: Pivot Optimiser [22]

การหาค่าที่ดีที่สุดนั้นเป็นหนึ่งในความท้าทายทางการวิจัยเนื่องจาก ไม่มีคำตอบใดที่เป็นคำตอบที่ดีที่สุดของปัญหาเหล่านี้ แนวคิดและอัลกอริทึมที่พัฒนาขึ้นมานั้นทำได้เพียงแค่หาคำตอบที่เหมาะสมที่สุดในการใช้งาน (Feasible solution) จากคำตอบทั้งหมดที่มีไม่รู้จบ (Real-number space boundary)

อัลกอริทึมเหล่านี้มีการจัดจำแนกได้หลายประเภท แต่หากจัดประเภทตามทิศทางของการค้นหาคำตอบนั้น อัลกอริทึมประเภท Meta-heuristic ได้รับความนิยมเป็นอย่างมากในงานด้านนี้ เนื่องจากปัจจัย 2 ประการด้วยกันคือ

- ความยืดหยุ่นในการประยุกต์ใช้งาน (Flexible implementation) เนื่องจากออกแบบมาสำหรับปัญหาที่เป็น black box problem หรือ ปัญหาใด ๆ ที่ไม่ขึ้นกับตัวแปรภายใน ทำให้ง่ายต่อการนำอัลกอริทึมที่พัฒนาขึ้นมาไปใช้งาน
- ความเรียบง่ายของการออกแบบ (Simple concept design) เนื่องจากอัลกอริทึมหมวดหมู่นี้จะยึดหลักในการไม่ใช้ความชันของปัญหา (Gradient-free optimiser) ทำให้ง่ายต่อการพัฒนาและเพิ่มขอบเขตของปัญหาที่รองรับ

เป็นที่รู้กันว่า อัลกอริทึมที่ดีนั้นจะให้คำตอบใกล้เคียงกับคำตอบที่ดีที่สุดเสมอ ทั้งนี้เป็นเหตุอันเนื่องมาจากความสมเหตุสมผล (Reasonable) ของอัลกอริทึม

จากการทบทวนวรรณกรรมทำให้ทราบถึงปัจจัยที่มีผลต่อประสิทธิภาพของอัลกอริทึม ก่อนที่อัลกอริทึมประเภท Meta-heuristic จะได้รับความนิยม อัลกอริทึมในกลุ่มที่มีชื่อว่า direct search มันคืออัลกอริทึมที่ไม่ใช้แนวคิดของประชากร (Non-population based optimiser) และเป็น

อัลกอริทึมแบบ deterministic ซึ่งจะกำหนดทิศทางของการค้นหาคำตอบแบบตายตัวโดยไม่ใช้การ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุ่ม ทำให้คาดเดาคำตอบที่ออกมาได้อย่างแน่นอนและเป็นการค้นหาที่ครอบคลุมทั้งปริภูมิการค้นหา ซึ่งได้แนวคิดมาจากทฤษฎีต่างกันเช่น การค้นหาโดยใช้สัดส่วนทองคำ (Golden section search) การค้นหาที่น้อยที่สุดของ (the Nelder–Mead method) และ Luus–Jaakola เป็นต้น อัลกอริทึมที่กล่าวมานั้น ไม่ได้มีประสิทธิภาพที่จะหาคำตอบที่ดีที่สุด (Best answer) แต่อย่างใด แต่มีความสามารถที่จะหาคำตอบได้ครอบคลุมทั้งปริภูมิการค้นหา ซึ่งจะได้คำตอบที่มีค่าเฉลี่ยออกมาค่อนข้างดี แต่เนื่องจากคำตอบที่ค่อนข้างแย่และความซับซ้อนทางคณิตศาสตร์ที่น้อยมากทำให้ อัลกอริทึมกลุ่มดังกล่าวไม่เป็นที่นิยมในปัจจุบัน

ความสำเร็จของอัลกอริทึมตระกูล direct search สรุปได้ว่าอัลกอริทึมนี้ได้เลือกใช้ต้นแบบหรือได้รับแรงบันดาลใจในทางที่ถูกต้องซึ่งนำมาสู่การประยุกต์ที่ได้ผลลัพธ์ที่น่าเชื่อถือได้และมีความสมเหตุสมผลเพียงพอ

การหาค่าที่เหมาะสมที่สุดแบบจุด ที่ผู้วิจัยได้พัฒนาขึ้นมาขึ้นมานั้นได้เลือกใช้ทฤษฎี ทางเรขาคณิต ซึ่งรองรับการคำนวณแบบหลายมิติ (Multidimensional geometric method) มาเป็นแนวคิด ต้นแบบของการพัฒนาอัลกอริทึมการหาค่าที่เหมาะสมที่สุดซึ่งจะประกอบไปด้วยวิธีการดังนี้

5.2 จุดหมุน (Pivot)

จุดหมุนหรือ Pivot เปรียบเสมือน 1 หน่วยย่อยในการค้นหาคำตอบ จุดหมุนจะมีอยู่ 2 ประเภท คือ จุดหมุนศักยภาพ (Potential Pivot) ทำหน้าที่เน้นไปที่การหาคำตอบที่ดีที่สุด และ จุดหมุนควบคุม (Control Pivot) ทำหน้าที่ควบคุมระยะการค้นหาของจุดหมุนอื่น จุดหมุนทั้ง 2 แบบ ทำงานโดยใช้กลไกเดียวกัน แต่แตกต่างกันที่จุดประสงค์

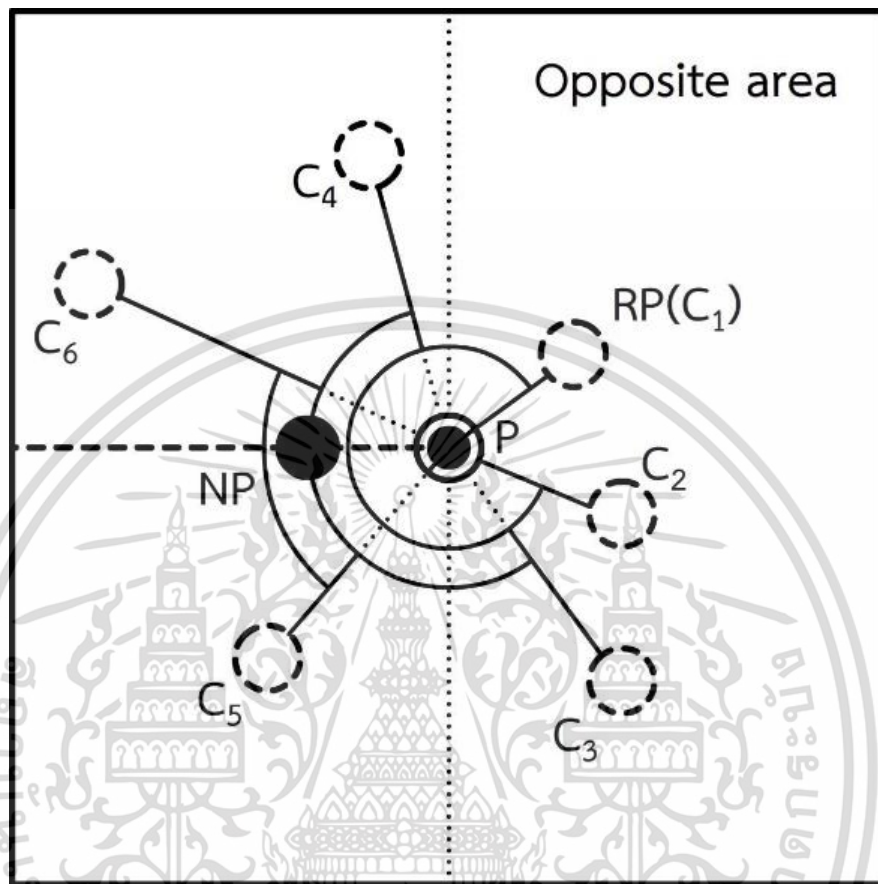
อัลกอริทึมนี้จะใช้อัลกอริทึมเชิงการวิวัฒนาการในการอธิบายโดย ให้ 1 จุดหมุนใด ๆ คือ ประชากร 1 ตัว ภายในจะประกอบไปด้วย เวกเตอร์การตัดสินใจ (Decision Vector) และคำตอบ พร้อมกับค่าเพิ่มเติมที่ใช้ในการคำนวณในอัลกอริทึมดังนี้

- รายการของระยะห่างเชิงเส้น (Linear distance list)
- รายการของระยะห่างเชิงมุม (Angular distance list)
- จุดหมุนที่ใกล้ที่สุด (Nearest pivot ,NP)
- จุดหมุนคู่แข่ง (Rival pivot ,RP)

รายการทั้ง 2 นั้น คือ ระยะทางเชิงเส้น (Euclidian distance) และระยะทางเชิงมุม (Angular distance) จากจุดหมุนที่พิจารณาไปยังจุดหมุนอื่น ๆ ทั้งหมด จุด NP นั้นคือจุดหมุนอื่นที่อยู่ใกล้เคียงกับจุดหมุนที่กำลังพิจารณาอยู่มากที่สุด คือจุดหมุนแรกในรายการระยะห่างเชิงเส้น

ในจุดนี้ เมื่อพิจารณาจุดหมุนใด ๆ จุดหมุนอื่นที่ไม่ใช่ NP จะถือว่าเป็นจุดหมุนรับเลือก (Candidate pivots ,CPs) ซึ่ง CP ที่อยู่ฝั่งตรงกันข้ามกับจุดหมุน (มีระยะห่างเชิงมุมมากกว่า 180 องศา หรือมากกว่า π) และมีระยะทางที่ใกล้ที่สุด จะได้รับเลือกให้เป็น RP รูปที่ 5.1 อธิบายถึงค่าตัว

แปรต่าง ๆ และวิธีการจำแนก NP CP และ RP โดยให้จุดหมุนทั้งหมดมีจำนวน 8 จุดและจุดหมุนที่พิจารณาคือจุดหมุน P



รูปที่ 5.1 พื้นที่จำลองในระนาบ 2 มิติ และตำแหน่งของ NP และ RP

ในบางกรณีเมื่อมิติของปัญหามีจำนวนมากขึ้น จะทำให้ไม่เข้าเงื่อนไขการหา RP ทั้ง 2 อย่าง จะเลือกให้จุดหมุนที่อยู่ห่างไกลจากจุด P ในระยะทางเชิงมุมเป็น RP

5.3 การสร้างประชากรใหม่ (Offspring Creation)

หลังจากการเลือกจุด RP จุดหมุนจะทำการหาคำตอบซึ่งเป็นการสร้างประชากรรุ่นใหม่ ออกมา กระบวนการสร้างประชากรใหม่จะทำการเรียงลำดับประชากรทั้งหมดโดยใช้คำตอบจาก คำตอบที่ดีที่สุดไปยังคำตอบที่แย่ที่สุด และแบ่งประชากรหรือจุดหมุนออกเป็น 2 หมวดหมู่ตาม คำตอบ ที่ควบคุมโดยค่าสัมประสิทธิ์ 2 ตัว คือ ค่าสัมประสิทธิ์ด้านล่าง (Lower factor ,LF) และ ค่า สัมประสิทธิ์ด้านบน (Upper factor ,UF)

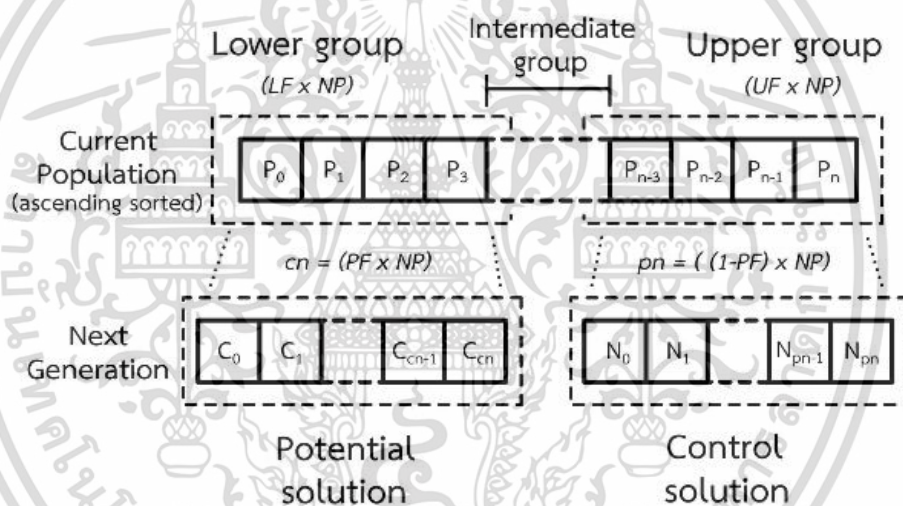
จุดหมุนที่มีคำตอบที่ดีที่สุดคิดเป็นสัดส่วนตาม LF จากประชากรทั้งหมดจะจัดให้เป็น กลุ่ม ล่าง (Lower Group) หรือเรียกว่า จุดหมุนศักยภาพ (Potential Pivot) ในขณะที่ จุดหมุนที่มีคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แยที่ที่สุดคิดเป็นสัดส่วนตาม UF จากประชากรทั้งหมดจะจัดให้เป็น กลุ่มบน (Upper Group) หรือเรียกว่า จุดหมุนควบคุม (Control Pivot) ซึ่ง LF และ PF จะมีค่าเริ่มต้นอยู่ที่ 0.25 สามารถปรับเปลี่ยนได้ตามจุดประสงค์ของการทำงาน จุดหมุนที่เหลือจะถูกเรียกว่า จุดหมุนระดับกลาง (Intermediate Pivot) เฉพาะจุดหมุน กลุ่มบนและกลุ่มล่างนั้นที่จะถูกใช้ในการสร้างประชากรในรุ่นต่อไป โดยการสุ่มแบบเอกรูปในประชากรกลุ่มล่างจำนวน cn ตัว และประชากรกลุ่มบนจำนวน pn โดยคำนวณจากสมการที่ 5.1

$$cn = PF \times NP, pn = (1 - PF) \times NP \quad (5.1)$$

เมื่อ NP คือจำนวนของประชากรทั้งหมด ประชากรรุ่นใหม่จะถูกสร้างมาเท่ากับประชากรในรุ่นเดิมดังแสดงในรูปที่ 5.2



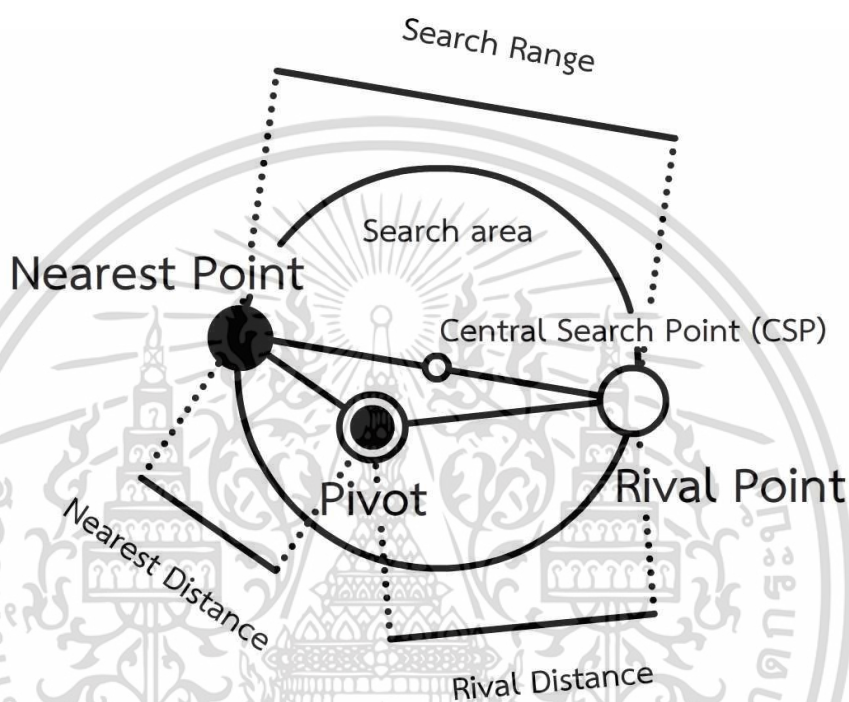
รูปที่ 5.2 จำนวนและประเภทของประชากรใหม่

ประชากรที่สร้างขึ้นใหม่จากจุดหมุนศักยภาพ (Potential Pivot) จะมีแนวโน้มไปในทิศทางที่ได้คำตอบที่ดีที่สุดเนื่องจาก เกิดจากจุดหมุนที่มีคำตอบที่ดีที่สุดส่วนใหญ่ผสมกัน แต่เนื่องจากมีเฉพาะจุดหมุนที่มีผลลัพธ์ที่ดีจึงเกิดเหตุการณ์การกระจุกตัวของคำตอบซึ่งทำให้ไม่สามารถออกจากจุดต่ำสุดสัมพัทธ์ (Local minimum) จึงต้องใช้จุดหมุนควบคุม (Control pivot) ซึ่งจะทำการสุ่มหาคำตอบที่แตกต่างออกไป จึงสามารถกระจายการหาคำตอบได้ทั่วทั้งปริภูมิการค้นหา (Search space)

สำหรับการหาคำตอบใหม่ของทุก ๆ จุดหมุน จะใช้จุดกึ่งกลางระหว่าง NP และ RP ของแต่ละจุดหมุนซึ่งจะให้ชื่อว่า จุดศูนย์กลางการค้นหา (Central Search Point , CSP) โดยคำนวณจากระยะห่างทางเรขาคณิตในทุกมิติ และคำนวณ พิสัยการค้นหา (search range) และทำการสุ่มด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้การกระจายแบบปกติ (Normal random distribution) ตามสมการการแจกแจงความน่าจะเป็นที่ (2.4) โดยกำหนดให้ μ มีค่าเท่ากับ CSP และ σ มีค่าเท่ากับ พิสัยการค้นหา ผลลัพธ์ที่ได้ ออกมานั้นจะมีแนวโน้มอธิบายดังรูปที่ 5.3 ซึ่งจำลองการหาค่าตอบใหม่ในระนาบ 2 มิติ ด้วยการกระจายตัวแบบปกติจะทำให้ค่าตอบนั้นมีความแน่นอนที่มากขึ้น แต่ยังมีคุณสมบัติของการสุ่มที่ทำให้มีโอกาสไปยังจุดต่ำสุดสัมพัทธ์ได้ง่ายขึ้น



รูปที่ 5.3 แนวโน้มของการหาค่าตอบใหม่

5.4 การคัดสรรประชากร (Selection)

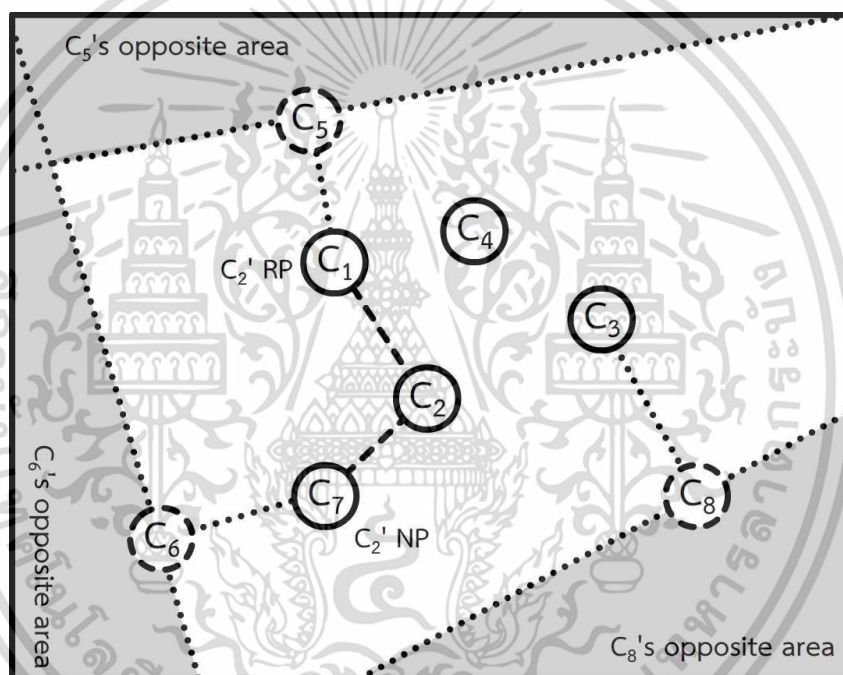
เมื่อการทำงานในขั้นตอนที่ 5.3 เสร็จสิ้นจะทำให้เกิดประชากรใหม่ขึ้นมาเมื่อรวมกับประชากรเดิมจะทำให้จำนวนประชากรเป็น 2 เท่าของที่มี จึงต้องทำการลบประชากรที่ไม่จำเป็นออกจากระบบ คุณสมบัติ 2 อย่างที่มีส่วนในการเลือกกำจัดประชากรคือ การเป็นจุดหมุนขอบ (Border pivots) และการเป็นจุดหมุนไม่ซ้ำซ้อน (Non-redundant pivots)

- จุดหมุนขอบ (Border pivots) คือจุดหมุนที่อยู่บริเวณขอบของกลุ่มโดยวัดจากระยะห่างเชิงมุม เมื่อพิจารณาจุดหมุนใดจุดหมุนหนึ่ง หากไม่มีจุดหมุนใด ๆ ที่อยู่ในพื้นที่ฝั่งตรงข้าม (Opposite area) จะถือว่าจุดหมุนนี้เป็นจุดหมุนที่อยู่ขอบ ซึ่งจุดหมุนนี้อาจจะให้ผลลัพธ์ที่ไม่ดีนักแต่เป็นจุดหมุนที่ควบคุมบริเวณการค้นหาของจุดหมุนอื่นได้

- จุดหมุนไม่ซ้ำซ้อน (Non-redundant pivots) หากจุดหมุนใด ๆ ที่อยู่ในบริเวณของกลุ่มจุดหมุนอื่นๆ แต่ให้ผลลัพธ์ที่แยกว่าจุดหมุนอื่นในกลุ่ม จะถือว่าเป็นจุดหมุนซ้ำซ้อน (Redundant pivots) หากจุดหมุนนั้นไม่ใช่จุดหมุนซ้ำซ้อนจะถือว่าเป็นจุดหมุน จุดหมุนไม่ซ้ำซ้อน

การจะหาคุณลักษณะดังกล่าวนี้จะทำการหา RP และ NP ใหม่อีกรอบแต่จะใช้การหาระยะห่างแบบ Mahalanobis (MD) ตามสมการที่ (2.2) แทนการหาระยะห่างแบบทั่วไป หรือ ยูคลิด (Euclidean space) ตามสมการที่ 2.1 ซึ่งจะให้ข้อมูลที่มีความแปรผันตามกลุ่มของประชากรมากกว่า การหาระยะห่างซึ่งจะให้ค่าแปรผันตามปริภูมิ

จุดหมุนซ้ำซ้อน คือจุดหมุนที่มีค่าคำตอบน้อยกว่าจุดหมุน NP และ RP ส่วนจุดหมุนขอบคือจุดหมุนที่ RP ไม่ได้ในพื้นที่ฝั่งตรงข้าม ตามรูปที่ 5.4



รูปที่ 5.4 การหาจุดหมุนขอบและจุดหมุนไม่ซ้ำซ้อน

รูปที่ 5.4 แสดงให้เห็นถึงตัวอย่างการคัดสรรโดยการหาจุดหมุนขอบและจุดหมุนไม่ซ้ำซ้อน เมื่อพิจารณาจุดหมุน C2 จะได้ว่าจุดหมุน C7 เป็น NP ของจุดหมุน C2 และจุดหมุน C1 เป็น RP ของจุดหมุน C2 หากคำตอบของ C2 นั้นมีค่าที่แยกว่า C7 และ C1 จะถือว่าเป็นจุดหมุนซ้ำซ้อน ในขณะที่จุดหมุน C5 C6 และ C8 ไม่มีจุดหมุนใด ๆ ที่ฝั่งตรงกันข้าม จะถือว่าเป็นจุดหมุน C5 C6 และ C8 เป็นจุดหมุนขอบ

หากจุดหมุนใด ๆ ในกลุ่มประชากรใหม่นั้น ไม่เป็นจุดหมุนขอบ และเป็นจุดหมุนซ้ำซ้อนจะถูกปล่อยออกจากประชากรทั้งหมดในระยะเวลาการทำงานนี้ กระบวนการคัดสรรจะถูกทำกับประชากรทั้งหมด ยกเว้นในประชากรกลุ่มล่างเพื่อให้หาคำตอบที่ดีที่สุดต่อไป

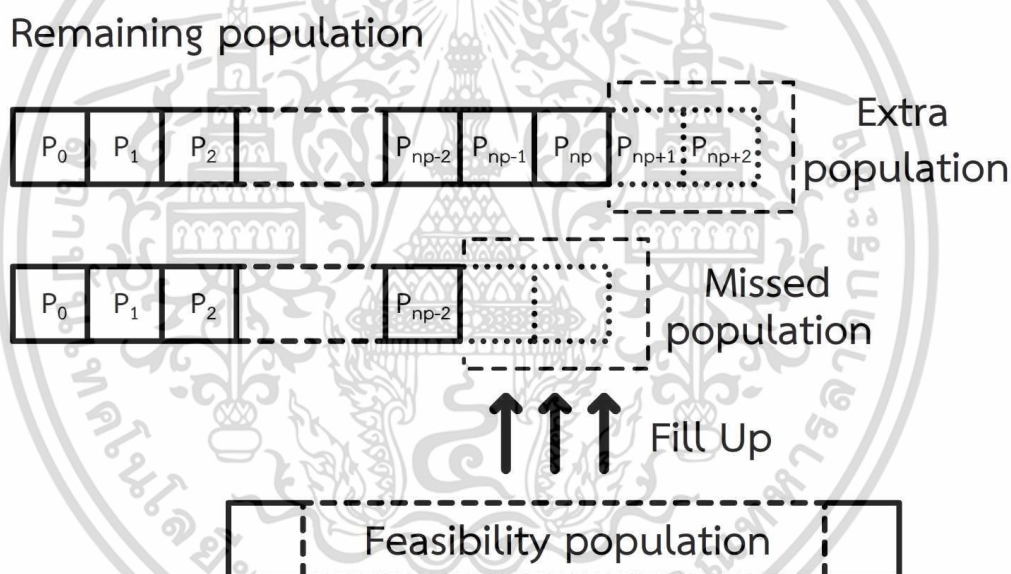
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 การชดเชยประชากร (Population Compensation)

หลังจากระยะเวลาทำงานของการคัดสรร เหตุการณ์ทั้งหมดที่เกิดขึ้นได้จะมี 2 แบบคือ จำนวนประชากรที่เหลืออยู่นั้นมีค่ามากกว่าและน้อยกว่าประชากรตั้งต้นการดำเนินการจะเป็นไปดังนี้ ตามรูปที่ 5.5

1) จำนวนประชากรที่เหลืออยู่มีค่ามากกว่าประชากรตั้งต้น หมายความว่าประชากรทั้งหมดมีประสิทธิภาพ และยังสามารถหาคำตอบที่ดีที่สุดโดยประชากรเดิมได้ ในกรณีนี้จะทำการลบประชากรที่ได้คำตอบแต่ที่เกินจากจำนวนประชากรตั้งต้นออกไป

2) จำนวนประชากรที่เหลืออยู่มีค่าน้อยกว่าประชากรตั้งต้น หมายความว่าพื้นที่ที่กำลังค้นหาอยู่นั้นมีศักยภาพการค้นหาน้อยอาจเกิดจากการตกอยู่ภายใต้จุดต่ำสุดสัมพัทธ์ ทำให้ต้องหาพื้นที่และประชากรตัวใหม่มาแทนที่ ในกรณีนี้จะทำการเติม จุดหมุนเข้าไปในประชากร โดยหามาจาก กลุ่มประชากรที่มีความเป็นไปได้ (Feasibility population)



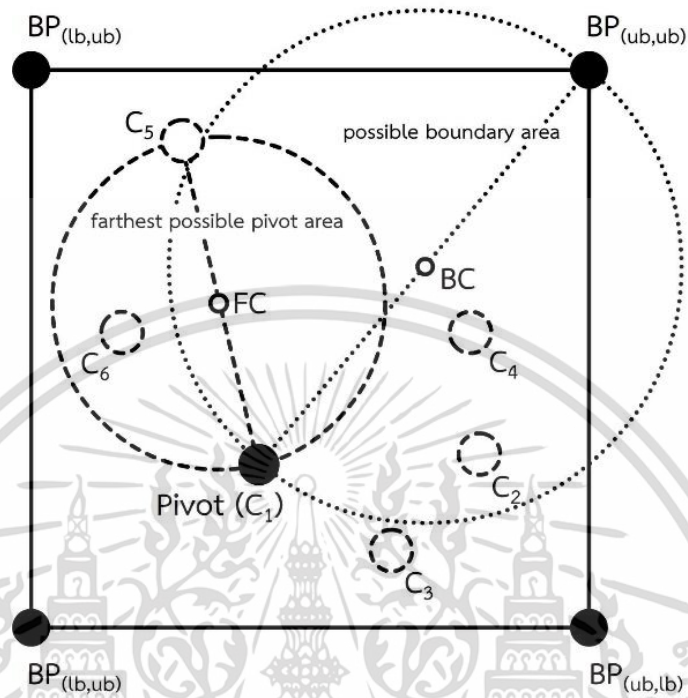
รูปที่ 5.5 การลบและเติมประชากรหลังจากการคัดสรร

Feasibility population หรือ กลุ่มของประชากรที่นำมาเติมนั้น จะสร้างมาจากคุณลักษณะ 2 อย่างด้วยกันคือ ขอบของการค้นหา (Boundary) สมมุติให้ทุกมุมของขอบการค้นหาในทุกมิติมีจุดหมุน จะให้จุดหมุนดังกล่าวที่มีระยะทางไกลที่สุดจากจุดหมุนที่พิจารณาเป็น จุดหมุนขอบ (Boundary Pivot, BP) BP นั้นจะใช้ในการกำหนดพื้นที่ใหม่ในการค้นหาคำตอบของกลุ่มประชากรใหม่ คุณลักษณะที่ 2 คือจุดหมุนที่ไกลที่สุด จากจุดหมุนที่พิจารณา (Farthest pivot, FP)

สุดท้ายจะหาจุดกึ่งกลางระหว่างจุดหมุนที่พิจารณา กับ BP เรียกว่า จุดกึ่งกลางขอบ (Centre point, BC) และจุดกึ่งกลางระหว่างจุดหมุนที่พิจารณา กับ FP เรียกว่า จุดกึ่งกลางไกลสุด (Farthest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pivot centre point ,FC) ตามรูปที่ 5.6 และสร้างประชากรใหม่จากจุดทั้ง 2 ด้วยวิธีการสุ่มแบบปกติ โดยให้ BP และ FP เป็น μ และระยะห่างระหว่างจุดเป็น σ เพื่อควบคุมพื้นที่ในการค้นหา



รูปที่ 5.6 การสร้างกลุ่มประชากรที่มีความเป็นไปได้

ประชากรกลุ่มใหม่ที่สร้างขึ้นมานี้จะถูกเติมเข้าไปในกลุ่มประชากรหลักตามค่าคำตอบที่มากที่สุดหากกลุ่มประชากรหลักเกิดการขาดแคลนประชากร

5.6 การทำงานของอัลกอริทึม (Process Flow)

การทำงานทั้งหมดจะถูกทำซ้ำไปเรื่อย ๆ โดยเป็นรอบการทำงานหรือ รุ่น (Generation) ตามอัลกอริทึมวิวัฒนาการตามรหัสเทียมในรูปที่ 5.7

```

0.) Prepare dataset from origin population

REPEAT UNTIL REQUIREMENT MET
1.) Prepare geometric data
    a. Build angular distance list
    b. Sort geometric data by distance
2.) Create offspring
    a. Select population
    b. Determine search range
    c. Merge new population
3.) Remove unnecessary population
4.) Adjust number of populations

```

รูปที่ 5.7 รหัสเทียมของการหาค่าเหมาะสมที่สุดแบบจุดหมุน

5.7 การทดลองประสิทธิภาพของ Pivot optimizer

เพื่อความเท่าเทียมของการทดลอง การทดลองการทำงานของอัลกอริทึมทั้งหมดจะทำอยู่บนสภาพแวดล้อมที่เรียกว่า pagmo platform [20] ซึ่งจะบรรจุปัญหาและอัลกอริทึมอื่นเพื่อทำการเปรียบเทียบประสิทธิภาพ รายการฟังก์ชันทดสอบทั้งหมดจะมีทั้งหมด 20 ฟังก์ชัน ซึ่งจะแบ่งเป็นฟังก์ชัน 4 ประเภท ได้แก่

- ฟังก์ชันมาตรฐานในการประเมินผลอัลกอริทึมการหาค่าเหมาะสมที่สุด (F1 ถึง F5)
- การหมุนและเพิ่มระดับฟังก์ชันมาตรฐาน (Shifted and Rotated) (F6 ถึง F11)
- ฟังก์ชันที่ใช้ใน CEC (F12 ถึง F16)
- ฟังก์ชันผสม (Hybrid Function) ของ ฟังก์ชัน CEC (F17 ถึง F20)

การทดลองจะทำโดยใช้ฟังก์ชันปัญหาในการหาค่าที่ดีที่สุด (Benchmark function) โดยกำหนดค่ารอบในการคำนวณคำตอบ (Function evaluation) ให้เท่ากันและกำหนดค่าพารามิเตอร์ต่างๆ ในแต่ละอัลกอริทึมให้เป็นค่าเริ่มต้น โดยฟังก์ชันที่ใช้ทดสอบและคุณสมบัติของแต่ละฟังก์ชันจะมีดังตารางที่ 5.1 นี้ ซึ่งสามารถดูรายละเอียดได้จากภาคผนวก ก. ฟังก์ชันทดสอบที่ใช้ในการทดลอง (Test Function)

ตารางที่ 5.1 ฟังก์ชันทดสอบที่ใช้ในการทดลองและคุณสมบัติ

No.	Function Name	Function Type	F*	Distribution
F1	Rosenbrock function	Basic function	0	Multimodal
F2	Rastrigin function	Basic function	0	Multimodal
F3	Schwefel function	Basic function	0	Multimodal
F4	Ackley function	Basic function	0	Multimodal
F5	Griewank function	Basic function	0	Multimodal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F6	Rotated Bent Cigar function (CEC'14 no. 2)	Shifted and Rotated Basic function	200	Unimodal
F7	Rotated Discus function (CEC'14 no. 3)	Shifted and Rotated Basic function	300	Unimodal
F8	Shifted and Rotated Rosenbrock function (CEC'14 no. 4)	Shifted and Rotated Basic function	400	Multimodal
F9	Shifted and Rotated Ackley function (CEC'14 no. 5)	Shifted and Rotated Basic function	500	Multimodal
F10	Shifted and Rotated Weierstrass function (CEC'14 no. 6)	Shifted and Rotated Basic function	600	Multimodal
F11	Shifted and Rotated Griewank function (CEC'14 no. 7)	Shifted and Rotated Basic function	700	Multimodal
F12	Shifted and Rotated Katsuura function	CEC (CEC'14 no. 12)	1200	Multimodal
F13	Shifted and Rotated HappyCat function	CEC (CEC'14 no. 13)	1300	Multimodal
F14	Shifted and Rotated HGBat function	CEC (CEC'14 no. 14)	1400	Multimodal
F15	Shifted and Rotated Expanded Griewank plus Rosenbrock function	CEC (CEC'14 no. 15)	1500	Multimodal
F16	Shifted and Rotated Expanded Scaffer's F6 function	CEC (CEC'14 no. 16)	1600	Multimodal
F17	CEC'14 no. 17 function	Hybrid function	1700	Multimodal
F18	CEC'14 no. 18 function	Hybrid function	1800	Multimodal
F19	CEC'14 no. 19 function	Hybrid function	1900	Multimodal
F20	CEC'14 no. 20 function	Hybrid function	2000	Multimodal

โดยการทดลองของแต่ละอัลกอริทึมนั้นจะทำการทดลองทั้งหมด 50 ครั้งโดยในแต่ละครั้งจะกำหนดให้เงื่อนไขการสิ้นสุดการทำงานคือรอบของการหาค่าตอบซึ่งถูกตั้งไว้ที่ $10000 \times D$ เมื่อ D คือมิติของปัญหา และเก็บผลทางสถิติค่าที่ดีที่สุด ค่าที่แย่ที่สุด ค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน ของการทดสอบ 50 ครั้งเพื่อการวิเคราะห์ผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.8 ผลการทดลอง

การทดลองจะนำฟังก์ชันทดสอบตั้งแต่ F1 ถึง F20 จำนวน 20 ฟังก์ชันตามตารางที่ 5.1 มาทดสอบในอัลกอริทึมที่แตกต่างกัน 5 อัลกอริทึมด้วยกันคือ อัลกอริทึมของผูู้วิจัย (Pivot optimization) PSO GWO DE และ ABC ในมิติของปัญหาที่แตกต่างกันคือ 10 50 และ 100 มิติ

ตารางที่ 5.2 ผลการทดลองกับฟังก์ชันทดสอบที่ 10 มิติ

Function		Pivot Optimiser	GWO	PSO	DE	ABC
F1 Rosenbrock	Avg.	5.7292E-03	5.9858E+00	1.3135E+00	4.0441E-05	2.7456E-01
	Best	6.3383E-15	5.1524E+00	6.3867E-03	1.0784E-05	1.2376E-02
	Worse	8.5647E-02	7.1882E+00	3.6045E+00	1.3127E-04	1.0711E+00
	SD	1.6549E-02	5.9297E-01	1.0582E+00	2.5668E-05	2.3789E-01
F2 Rastrigin	Avg.	4.5588E-09	7.3896E-15	2.7004E+00	2.9678E-06	6.0045E-10
	Best	2.0080E-11	0.0000E+00	0.0000E+00	6.8727E-08	0.0000E+00
	Worse	1.4631E-08	2.8422E-14	4.9748E+00	2.0322E-05	1.2175E-08
	SD	3.5535E-09	7.6475E-15	1.2373E+00	4.5325E-06	2.1920E-09
F3 Schwefel	Avg.	7.2366E-09	1.3050E+03	2.8900E+02	1.6342E-07	4.6907E+00
	Best	4.7294E-11	6.9090E+02	0.0000E+00	6.0457E-08	9.0949E-13
	Worse	2.4949E-08	1.9998E+03	5.9371E+02	2.6635E-07	1.1844E+02
	SD	5.3795E-09	3.0627E+02	1.6176E+02	4.5199E-08	2.0894E+01
F4 Ackley	Avg.	6.8360E-08	7.4074E-15	4.1389E-15	6.3212E-07	1.1315E-14
	Best	1.7640E-08	3.9968E-15	3.9968E-15	3.5628E-07	7.5495E-15
	Worse	1.2700E-07	7.5495E-15	7.5495E-15	8.5835E-07	1.4655E-14
	SD	3.0899E-08	6.9619E-16	6.9619E-16	1.2321E-07	2.7893E-15
F5 Griewank	Avg.	5.0765E-12	1.0877E-02	2.2847E-02	7.1928E-02	2.2412E-03
	Best	0.0000E+00	0.0000E+00	0.0000E+00	2.7219E-02	0.0000E+00
	Worse	2.3440E-10	9.2053E-02	5.4128E-02	1.1222E-01	2.4610E-02
	SD	3.2789E-11	1.7833E-02	1.2965E-02	1.9347E-02	4.9757E-03
F6 Bent Cigar	Avg.	2.0000E+02	5.4081E+06	6.8839E+02	2.0049E+02	8.5087E+02
	Best	2.0000E+02	4.2992E+02	2.0004E+02	2.0014E+02	2.2695E+02
	Worse	2.0000E+02	2.7023E+08	5.6298E+03	2.0112E+02	1.9800E+03
	SD	2.5710E-10	3.7832E+07	9.3707E+02	1.9407E-01	4.0925E+02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F7 Discus	Avg.	3.0000E+02	4.3355E+03	6.2287E+02	3.0000E+02	6.1826E+03
	Best	3.0000E+02	3.3874E+02	3.0418E+02	3.0000E+02	2.2996E+03
	Worse	3.0000E+02	1.7239E+04	1.8344E+03	3.0000E+02	9.8172E+03
	SD	7.9688E-11	4.2702E+03	3.7844E+02	1.3995E-07	1.3426E+03
F8 Rosenbrock	Avg.	4.0000E+02	4.3121E+02	4.0535E+02	4.0026E+02	4.0025E+02
	Best	4.0000E+02	4.0051E+02	4.0002E+02	4.0000E+02	4.0005E+02
	Worse	4.0000E+02	4.3834E+02	4.3478E+02	4.0434E+02	4.0091E+02
	SD	1.5420E-04	1.0256E+01	9.5411E+00	1.0294E+00	1.7094E-01
F9 Ackley	Avg.	5.1632E+02	5.2170E+02	5.1815E+02	5.1996E+02	5.1967E+02
	Best	5.0382E+02	5.2108E+02	5.0000E+02	5.1573E+02	5.0170E+02
	Worse	5.2009E+02	5.2213E+02	5.2034E+02	5.2024E+02	5.2118E+02
	SD	5.7312E+00	2.3668E-01	5.9085E+00	7.5289E-01	3.3322E+00
F10 Weierstrass	Avg.	6.0001E+02	6.0208E+02	6.0029E+02	6.0011E+02	6.0438E+02
	Best	6.0000E+02	6.0011E+02	6.0000E+02	6.0001E+02	6.0182E+02
	Worse	6.0006E+02	6.0564E+02	6.0185E+02	6.0037E+02	6.1302E+02
	SD	1.4682E-02	1.0807E+00	4.0440E-01	8.9224E-02	1.9660E+00
F11 Griewank	Avg.	7.0001E+02	7.0135E+02	7.0003E+02	7.0022E+02	7.0002E+02
	Best	7.0000E+02	7.0008E+02	7.0000E+02	7.0006E+02	7.0000E+02
	Worse	7.0003E+02	7.0419E+02	7.0007E+02	7.0029E+02	7.0006E+02
	SD	7.3348E-03	1.0658E+00	1.4857E-02	4.6784E-02	1.2227E-02
F12 Katsuura	Avg.	1.2003E+03	1.2074E+03	1.2001E+03	1.2005E+03	1.2008E+03
	Best	1.2002E+03	1.2000E+03	1.2000E+03	1.2004E+03	1.2001E+03
	Worse	1.2004E+03	1.2297E+03	1.2002E+03	1.2007E+03	1.2046E+03
	SD	5.1528E-02	8.3995E+00	4.6009E-02	8.4892E-02	9.2322E-01
F13 HappyCat	Avg.	1.3001E+03	1.3003E+03	1.3001E+03	1.3002E+03	1.3037E+03
	Best	1.3001E+03	1.3001E+03	1.3000E+03	1.3001E+03	1.3003E+03
	Worse	1.3002E+03	1.3007E+03	1.3002E+03	1.3003E+03	1.3064E+03
	SD	2.1793E-02	1.6579E-01	3.6887E-02	3.5739E-02	1.9507E+00
F14 HGBat	Avg.	1.4001E+03	1.4004E+03	1.4002E+03	1.4002E+03	1.4416E+03
	Best	1.4001E+03	1.4001E+03	1.4001E+03	1.4001E+03	1.4003E+03
	Worse	1.4002E+03	1.4018E+03	1.4003E+03	1.4002E+03	1.4808E+03
	SD	2.8371E-02	3.5080E-01	4.3127E-02	3.0146E-02	2.3859E+01
F15 Griewank & Rosenbrock	Avg.	1.5007E+03	1.5037E+03	1.5006E+03	1.5017E+03	7.9178E+04
	Best	1.5004E+03	1.5004E+03	1.5003E+03	1.5011E+03	1.5014E+03
	Worse	1.5009E+03	1.5148E+03	1.5008E+03	1.5021E+03	6.0401E+05
	SD	1.1977E-01	4.2191E+00	1.2296E-01	2.6338E-01	1.1863E+05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F16 Expanded Scaffer's F6	Avg.	1.6019E+03	1.6031E+03	1.6020E+03	1.6026E+03	1.6027E+03
	Best	1.6014E+03	1.6008E+03	1.6012E+03	1.6021E+03	1.6016E+03
	Worse	1.6023E+03	1.6059E+03	1.6026E+03	1.6029E+03	1.6033E+03
	SD	2.0383E-01	1.2523E+00	3.2509E-01	1.6484E-01	3.7179E-01
F17 CEC17	Avg.	1.7348E+03	1.3436E+04	3.0312E+03	1.7510E+03	5.4765E+05
	Best	1.7050E+03	2.7259E+03	1.8913E+03	1.7215E+03	3.7170E+04
	Worse	1.7709E+03	3.3612E+05	6.7436E+03	1.7849E+03	2.1654E+06
	SD	1.4644E+01	4.6216E+04	9.0305E+02	1.3952E+01	4.1687E+05
F18 CEC18	Avg.	1.8018E+03	1.1360E+04	1.9971E+03	1.8048E+03	2.1006E+04
	Best	1.8005E+03	1.9102E+03	1.8149E+03	1.8030E+03	3.5184E+03
	Worse	1.8040E+03	3.2892E+04	3.0844E+03	1.8075E+03	9.2947E+04
	SD	7.5285E-01	7.8243E+03	2.4845E+02	9.1941E-01	1.3939E+04
F19 CEC19	Avg.	1.9002E+03	1.9022E+03	1.9010E+03	1.9008E+03	1.9014E+03
	Best	1.9001E+03	1.9011E+03	1.9002E+03	1.9002E+03	1.9004E+03
	Worse	1.9004E+03	1.9036E+03	1.9016E+03	1.9012E+03	1.9041E+03
	SD	5.8293E-02	7.0947E-01	2.9534E-01	1.8397E-01	6.3970E-01
F20 CEC20	Avg.	2.0006E+03	5.9046E+03	2.2352E+03	2.0022E+03	7.0760E+03
	Best	2.0002E+03	2.0557E+03	2.0054E+03	2.0013E+03	2.1201E+03
	Worse	2.0012E+03	1.9585E+04	3.2860E+03	2.0035E+03	2.0112E+04
	SD	2.2798E-01	4.0823E+03	2.7177E+02	5.1551E-01	4.1968E+03

ตารางที่ 5.3 ผลการทดลองกับฟังก์ชันทดสอบที่ 50 มิติ

Function		Pivot Optimiser	GWO	PSO	DE	ABC
F1 Rosenbrock	Avg.	2.757658E+01	4.593013E+01	3.847835E+01	6.832442E+01	1.254850E-01
	Best	3.194159E+00	4.401039E+01	4.353736E+00	6.707065E+01	8.079473E-03
	Worse	3.888973E+01	4.837887E+01	9.010989E+01	6.944334E+01	6.540980E-01
	SD	6.497239E+00	8.415640E-01	1.939872E+01	5.479016E-01	1.485786E-01
F2 Rastrigin	Avg.	6.857309E-08	3.865352E-14	1.078138E+02	1.699961E-04	2.168674E+02
	Best	1.736083E-08	0.000000E+00	7.064202E+01	5.088959E-05	7.189253E-07
	Worse	1.829657E-07	1.136868E-13	1.422786E+02	3.585511E-04	8.034852E+02
	SD	3.340798E-08	3.673334E-14	1.950924E+01	7.532920E-05	3.351617E+02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F3 Schwefel	Avg.	6.390437E-08	1.126131E+04	5.371383E+03	1.282216E-06	3.651531E+03
	Best	7.061317E-09	9.843771E+03	3.776413E+03	9.994765E-07	1.184383E+02
	Worse	1.338231E-07	1.276007E+04	6.761855E+03	1.575863E-06	1.815269E+04
	SD	2.797356E-08	7.766986E+02	6.472225E+02	1.509964E-07	6.472330E+03
F4 Ackley	Avg.	3.411650E-07	1.259437E-14	1.493916E-14	3.249790E-06	1.022148E+01
	Best	1.667049E-07	7.549517E-15	7.549517E-15	2.385096E-06	2.988720E-13
	Worse	5.444244E-07	1.820766E-14	3.597123E-14	3.979251E-06	2.007054E+01
	SD	9.106606E-08	2.850153E-15	4.133380E-15	4.351654E-07	9.539133E+00
F5 Griewank	Avg.	6.862080E-08	5.995204E-17	0.000000E+00	1.172477E-06	8.700308E+02
	Best	5.566453E-10	0.000000E+00	0.000000E+00	6.993345E-07	1.089830E+00
	Worse	2.309848E-07	2.220446E-16	0.000000E+00	1.501460E-06	1.167499E+03
	SD	4.351572E-08	7.095011E-17	0.000000E+00	1.926274E-07	3.989539E+02
F6 Bent Cigar	Avg.	2.000007E+02	5.139683E+09	2.616084E+03	1.061260E+05	1.674424E+04
	Best	2.000000E+02	1.608134E+09	2.002357E+02	4.532076E+04	3.136352E+03
	Worse	2.000106E+02	1.243813E+10	2.013893E+04	1.839038E+05	3.896929E+04
	SD	1.567619E-03	2.344273E+09	4.763273E+03	3.580030E+04	1.014004E+04
F7 Discus	Avg.	3.000000E+02	5.718613E+04	4.781180E+03	9.144959E+03	3.115976E+04
	Best	3.000000E+02	3.806963E+04	7.551119E+02	4.644820E+03	8.700327E+03
	Worse	3.000001E+02	8.703967E+04	9.240398E+03	1.644960E+04	4.614601E+04
	SD	1.066193E-05	1.143646E+04	2.001005E+03	2.228517E+03	6.429579E+03
F8 Rosenbrock	Avg.	4.631127E+02	9.097278E+02	4.787130E+02	7.832027E+02	4.916021E+02
	Best	4.000157E+02	6.536024E+02	4.226406E+02	7.456083E+02	4.302933E+02
	Worse	4.981031E+02	1.883599E+03	4.984787E+02	8.203847E+02	5.262534E+02
	SD	3.121517E+01	2.192333E+02	2.256567E+01	1.727374E+01	1.783916E+01
F9 Ackley	Avg.	5.203804E+02	5.217117E+02	5.206624E+02	5.208304E+02	5.213117E+02
	Best	5.203063E+02	5.214724E+02	5.202315E+02	5.207493E+02	5.205140E+02
	Worse	5.204269E+02	5.218905E+02	5.209507E+02	5.208731E+02	5.215394E+02
	SD	2.815029E-02	9.193074E-02	2.061204E-01	3.052135E-02	2.978641E-01
F10 Weierstrass	Avg.	6.265095E+02	6.289703E+02	6.358566E+02	6.916173E+02	6.657972E+02
	Best	6.220676E+02	6.207514E+02	6.277485E+02	6.845215E+02	6.362065E+02
	Worse	6.309385E+02	6.376631E+02	6.433710E+02	6.963402E+02	6.882450E+02
	SD	2.110837E+00	4.245630E+00	3.472582E+00	2.331359E+00	1.945706E+01
F11 Griewank	Avg.	7.000000E+02	7.438880E+02	7.000004E+02	7.000300E+02	3.072224E+03
	Best	7.000000E+02	7.093829E+02	7.000000E+02	7.000110E+02	7.010972E+02
	Worse	7.000007E+02	8.048245E+02	7.000074E+02	7.000462E+02	3.878383E+03
	SD	1.188365E-04	2.187597E+01	1.756462E-03	7.751223E-03	8.456054E+02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F12 Katsuura	Avg.	1.200350E+03	1.209321E+03	1.200578E+03	1.201089E+03	1.205274E+03
	Best	1.200272E+03	1.200051E+03	1.200209E+03	1.200867E+03	1.200417E+03
	Worse	1.200420E+03	1.218728E+03	1.201006E+03	1.201258E+03	1.208237E+03
	SD	3.640713E-02	5.977303E+00	1.266216E-01	8.098679E-02	2.662505E+00
F13 HappyCat	Avg.	1.300362E+03	1.300943E+03	1.300443E+03	1.300595E+03	1.312060E+03
	Best	1.300239E+03	1.300457E+03	1.300297E+03	1.300479E+03	1.300799E+03
	Worse	1.300434E+03	1.301819E+03	1.300628E+03	1.300664E+03	1.314242E+03
	SD	4.078151E-02	2.963927E-01	6.085019E-02	3.871812E-02	2.558386E+00
F14 HGBat	Avg.	1.400261E+03	1.408509E+03	1.400303E+03	1.400306E+03	1.986651E+03
	Best	1.400207E+03	1.400459E+03	1.400239E+03	1.400266E+03	1.400423E+03
	Worse	1.400311E+03	1.422545E+03	1.400353E+03	1.400346E+03	2.240003E+03
	SD	2.547386E-02	7.186517E+00	2.624065E-02	1.697790E-02	2.449531E+02
F15 Griewank & Rosenbrock	Avg.	1.511988E+03	2.288524E+03	1.514453E+03	1.574568E+03	1.171587E+08
	Best	1.508896E+03	1.525593E+03	1.509037E+03	1.566855E+03	1.566025E+03
	Worse	1.514610E+03	1.215279E+04	1.521032E+03	1.579550E+03	3.660834E+08
	SD	1.329203E+00	1.653862E+03	2.601225E+00	2.731169E+00	8.092458E+07
F16 Expanded Scaffer's F6	Avg.	1.617913E+03	1.621893E+03	1.619843E+03	1.642358E+03	1.622242E+03
	Best	1.617280E+03	1.618328E+03	1.618373E+03	1.640700E+03	1.619512E+03
	Worse	1.618412E+03	1.626296E+03	1.620966E+03	1.642905E+03	1.624423E+03
	SD	2.852451E-01	2.565994E+00	5.702727E-01	3.846455E-01	1.692080E+00
F17 CEC17	Avg.	1.697586E+04	3.086572E+06	3.878155E+05	3.746470E+07	6.346980E+07
	Best	6.781968E+03	2.464909E+05	1.016691E+05	2.108578E+07	1.140468E+07
	Worse	5.072359E+04	1.280370E+07	7.742536E+05	5.802825E+07	1.949951E+09
	SD	7.406777E+03	2.375746E+06	1.633254E+05	7.409549E+06	2.698458E+08
F18 CEC18	Avg.	2.042677E+03	4.702377E+07	2.518943E+03	8.550081E+04	7.785928E+09
	Best	1.852111E+03	2.706100E+03	2.092669E+03	3.427193E+04	1.984714E+05
	Worse	2.899351E+03	6.726308E+08	4.753744E+03	1.715983E+05	4.376088E+10
	SD	2.352941E+02	1.173142E+08	5.331499E+02	3.210232E+04	1.268599E+10
F19 CEC19	Avg.	1.914016E+03	1.976707E+03	1.918574E+03	2.007890E+03	3.675153E+03
	Best	1.909496E+03	1.938782E+03	1.913603E+03	1.987736E+03	1.928711E+03
	Worse	1.921285E+03	2.030684E+03	1.926432E+03	2.022749E+03	8.421961E+03
	SD	2.285185E+00	2.252713E+01	2.657467E+00	9.400473E+00	2.438670E+03
F20 CEC20	Avg.	2.066225E+03	1.754448E+04	6.544531E+03	4.637847E+04	1.581747E+07
	Best	2.033006E+03	6.671582E+03	3.436307E+03	3.029968E+04	7.954806E+04
	Worse	2.104327E+03	3.177888E+04	1.200669E+04	6.765575E+04	7.069587E+07
	SD	1.306982E+01	5.785352E+03	1.699718E+03	7.212320E+03	1.571466E+07

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.4 ผลการทดลองกับฟังก์ชันทดสอบที่ 100 มิติ

Function		Pivot Optimiser	GWO	PSO	DE	ABC
F1 Rosenbrock	Avg.	5.289565E+01	9.638721E+01	1.302354E+02	6.832442E+01	4.860048E-01
	Best	4.912656E-08	9.413218E+01	4.355201E+01	6.707065E+01	2.668739E-02
	Worse	8.794031E+01	9.829681E+01	2.270706E+02	6.944334E+01	2.410936E+00
	SD	3.591368E+01	1.078160E+00	3.201220E+01	5.479016E-01	5.065154E-01
F2 Rastrigin	Avg.	1.779719E-07	9.777068E-14	3.137502E+02	1.699961E-04	1.020528E+03
	Best	8.786230E-08	0.000000E+00	2.537228E+02	5.088959E-05	1.119429E+00
	Worse	3.435487E-07	2.273737E-13	3.860422E+02	3.585511E-04	1.658987E+03
	SD	4.760268E-08	7.544559E-14	3.266588E+01	7.532920E-05	7.314127E+02
F3 Schwefel	Avg.	1.845739E-07	2.519577E+04	1.250371E+04	1.282216E-06	2.976481E+04
	Best	7.072231E-08	2.124346E+04	9.985411E+03	9.994765E-07	2.969423E+03
	Worse	3.846944E-07	4.002007E+04	1.469986E+04	1.575863E-06	3.875570E+04
	SD	7.197216E-08	2.595181E+03	9.338137E+02	1.509964E-07	1.318363E+04
F4 Ackley	Avg.	6.354415E-07	1.678657E-14	8.962435E-01	3.249790E-06	1.751874E+01
	Best	2.983451E-07	1.110223E-14	5.018208E-14	2.385096E-06	8.996684E-01
	Worse	1.024654E-06	2.531308E-14	2.371859E+00	3.979251E-06	2.045016E+01
	SD	1.476404E-07	2.561898E-15	8.550489E-01	4.351654E-07	6.109071E+00
F5 Griewank	Avg.	2.088668E-07	5.107026E-17	0.000000E+00	1.172477E-06	1.774109E+03
	Best	6.611281E-08	0.000000E+00	0.000000E+00	6.993345E-07	1.239751E+00
	Worse	4.490511E-07	2.220446E-16	0.000000E+00	1.501460E-06	2.519229E+03
	SD	9.251611E-08	6.362266E-17	0.000000E+00	1.926274E-07	1.000456E+03
F6 Bent Cigar	Avg.	2.000000E+02	3.298789E+10	6.975116E+03	1.061260E+05	1.041441E+05
	Best	2.000000E+02	1.632013E+10	2.023611E+02	4.532076E+04	2.707895E+04
	Worse	2.000002E+02	5.125640E+10	3.995315E+04	1.839038E+05	2.198826E+05
	SD	3.209215E-05	8.195167E+09	8.526513E+03	3.580030E+04	4.426053E+04
F7 Discus	Avg.	3.000000E+02	1.561426E+05	4.378336E+03	9.144959E+03	5.733499E+04
	Best	3.000000E+02	9.740740E+04	1.421401E+03	4.644820E+03	3.091619E+04
	Worse	3.000001E+02	2.258403E+05	7.924109E+03	1.644960E+04	9.565370E+04
	SD	2.718126E-05	2.608683E+04	1.548991E+03	2.228517E+03	9.585855E+03
F8 Rosenbrock	Avg.	5.654782E+02	2.694468E+03	6.071386E+02	7.832027E+02	6.330870E+02
	Best	4.881237E+02	1.508411E+03	5.461348E+02	7.456083E+02	5.573579E+02
	Worse	6.283701E+02	5.096082E+03	6.845874E+02	8.203847E+02	7.056281E+02
	SD	3.493488E+01	6.650299E+02	2.843625E+01	1.727374E+01	3.092165E+01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F18 CEC18	Avg.	2.522794E+03	4.667303E+08	3.109273E+03	8.550081E+04	3.641937E+10
	Best	1.939620E+03	2.301620E+06	2.482861E+03	3.427193E+04	8.066706E+06
	Worse	4.411072E+03	1.278797E+09	5.494201E+03	1.715983E+05	9.766149E+10
	SD	4.866846E+02	2.768084E+08	6.326468E+02	3.210232E+04	4.098991E+10
F19 CEC19	Avg.	1.967645E+03	2.199877E+03	2.002385E+03	2.007890E+03	1.162825E+04
	Best	1.918337E+03	2.076907E+03	1.969781E+03	1.987736E+03	1.988721E+03
	Worse	2.006737E+03	2.365879E+03	2.013405E+03	2.022749E+03	3.305838E+04
	SD	1.717160E+01	6.565446E+01	1.220356E+01	9.400473E+00	1.113420E+04
F20 CEC20	Avg.	2.397155E+03	5.707377E+04	1.323976E+04	4.637847E+04	4.091894E+07
	Best	2.214549E+03	2.226062E+04	5.663089E+03	3.029968E+04	1.858900E+05
	Worse	2.724933E+03	1.068211E+05	2.217427E+04	6.765575E+04	1.365339E+08
	SD	1.029242E+02	1.949632E+04	3.278553E+03	7.212320E+03	3.082750E+07

5.9 การวิเคราะห์ผลการทดลอง

ในปัญหา 10 มิติ นั้น สถิติที่ดีที่สุดของแต่ละอัลกอริทึม นั้น ไม่ต่างกันนัก ค่าตอบของฟังก์ชันพื้นฐานและฟังก์ชันที่ทำการหมุน (F1 ถึง F16) ไม่ได้แสดงให้เห็นถึงความแตกต่างของแต่ละอัลกอริทึมมากนัก เห็นได้จากค่าที่ดีที่สุดจะสลับกันไปมาระหว่างแต่ละอัลกอริทึม ในด้านการหาคำตอบที่ดีที่สุดหรือ best solution นั้น PSO GWO และ ABC สามารถไปยังคำตอบที่ดีที่สุดได้หลายครั้งกว่า Pivot optimizer แต่ค่าเฉลี่ยนั้น ยังใกล้เคียงกัน ส่วนในฟังก์ชันผสมซึ่งมีความซับซ้อนมากขึ้น (F17 – F20) Pivot optimizer ได้ผลลัพธ์ที่ดีที่สุดเป็นส่วนใหญ่แต่ยังไม่มี ความแตกต่างกับอัลกอริทึมอื่นมากนัก จึงยังไม่สามารถสรุปประสิทธิภาพของอัลกอริทึมได้อย่างละเอียด

ส่วนปัญหา 50 มิติ นั้น เนื่องจากการเพิ่มมิติของปัญหาและรอบในการประมวลผลขึ้นหลายเท่า ผลที่ออกมาจึงแสดงความเด่นของแต่ละอัลกอริทึมออกมาชัดเจนมากขึ้น PSO และ GWO นั้นเป็นอัลกอริทึมที่สามารถไปยังคำตอบที่ดีที่สุดได้มากที่สุด โดยเฉพาะปัญหาที่ F1 – F16 ยกเว้นปัญหาที่มีลักษณะของฐานนิยมเดียว (unimodal) นั้น Pivot optimizer ได้ผลลัพธ์ออกมาดีกว่าในทั้งทางของค่าเบี่ยงเบนมาตรฐานและค่าที่ดีที่สุด และในปัญหาที่ F17 – 20 นั้น Pivot optimizer มีผลลัพธ์ที่ดีที่สุดในทุกด้าน

ปัญหาแบบ 100 มิติ ซึ่งเป็นปัญหาที่มีความยากที่สุดและใช้จำนวนรอบมากที่สุดนั้น ผลออกมาคล้ายกับปัญหา แบบ 50 มิติ โดยในปัญหาที่ซับซ้อน (F17 – F20) นั้น Pivot optimizer ได้ผลลัพธ์ที่ดีที่สุดเป็นส่วนใหญ่ แต่เมื่อเทียบกับปัญหา 50 มิติแล้ว มีประสิทธิภาพลดลงเมื่อเทียบกับอัลกอริทึมอื่น จะเห็นได้จากการที่สถิติที่ดีที่สุดนั้นเกิดการกระจายมากขึ้น และ PSO และ GWO

สามารถหาผลลัพธ์ที่ดีที่สุดได้แย่ง ส่วน DE มีค่า SD ที่น้อยลงอย่างมาก เมื่อเทียบกับอัลกอริทึมอื่น ใน ปัญหาที่มีมิติมาก

5.10 สรุปผลการทดลอง

จากผลการทดลองปรากฏว่า Pivot optimizer สามารถให้ค่าเฉลี่ยที่ดีที่สุดออกมาเมื่อเทียบกับอัลกอริทึมอื่น แต่ค่าเบี่ยงเบนมาตรฐานหรือ SD นั้นยังออกมาใกล้เคียงกับ DE และการหาค่าที่ดีที่สุดนั้นยังคงเป็นรอง PSO และ GWO โดยเฉพาะปัญหาที่ไม่ซับซ้อน

Pivot optimizer เป็นอัลกอริทึมที่ให้ผลลัพธ์ที่เฉลี่ยออกมาแล้วดีกว่าอัลกอริทึมอื่น โดยเฉพาะในปัญหาที่มีความซับซ้อนสูงและปัญหาที่มีมิติจำนวนมาก แต่ความเสถียรนั้นเมื่อดูจากค่า SD แล้วยังให้ค่าที่กระจายมากกว่า อัลกอริทึม DE ซึ่งอาจจะเกิดจากการที่ DE ใช้หลักการทางพันธุกรรม จึงให้ผลที่ใกล้เคียงกันในทุกรอบการประมวลผล และมีความสามารถในการหาจุดที่ดีที่สุดของปัญหาเป็นรอง GWO และ PSO

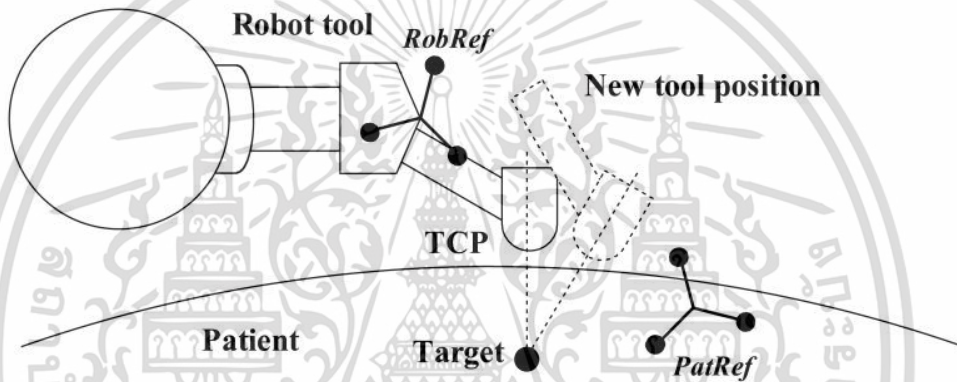


บทที่ 6

การประยุกต์ใช้งานคำนวณเส้นทางการเคลื่อนที่

6.1 บทนำ

หลังจากการทดสอบประสิทธิภาพของอัลกอริทึมด้วยฟังก์ชัน การทดสอบทั่วไปแล้ว บทนี้จะกล่าวถึงตัวอย่างการนำอัลกอริทึม Pivot optimizer ไปใช้งานจริงในงานทางวิศวกรรมที่เรียกว่า PTP motion (Point-to-point motion) หรือการเคลื่อนที่ระหว่างจุดของ Tool Center Point (TCP) หรือจุดกลางของเครื่องมือในเครื่องจักร

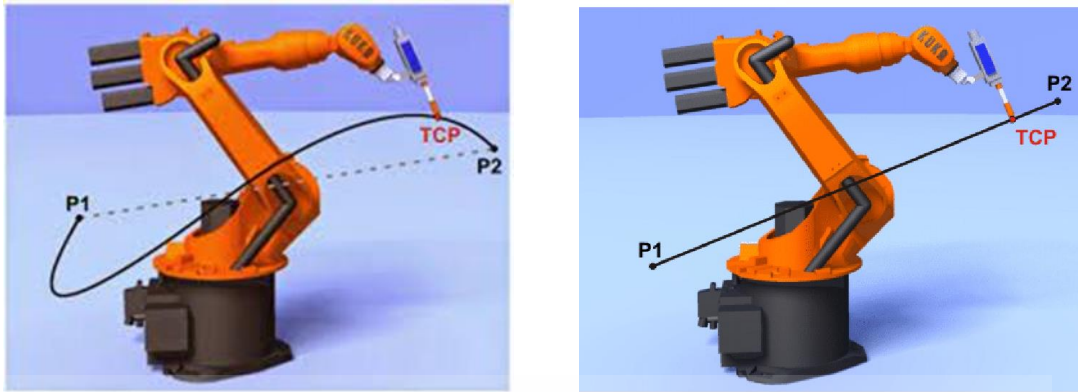


รูปที่ 6.1 จุดกลางเครื่องมือ (TCP) ของเครื่องจักร

ปัจจุบันในภาคอุตสาหกรรม ได้มีการประยุกต์ใช้เครื่องจักรที่ทำงานในรูปแบบอัตโนมัติ มาแทนที่เครื่องจักรในรูปแบบเก่าซึ่งใช้มนุษย์ในการควบคุมทั้งหมด เครื่องจักรดังกล่าวนี้มีชื่อประเภทที่เรียกว่า Computer Numerical Control หรือ CNC ซึ่งจะใช้หลัก คำนวณเส้นทางการเคลื่อนที่ที่ติดตั้งลงในเครื่องจักรไว้ล่วงหน้า แล้วจึงป้อนคำสั่งหรือ motion ลงในเครื่องจักรเพื่อให้ เครื่องจักรดังกล่าวทำงานตามที่ต้องการ ตัวอย่างเช่น หุ่นยนต์หยิบและวาง (Pick and place robot) เครื่องกัด (Machining center CNC) เครื่องตัดด้วยเลเซอร์ (Laser cut machine) เป็นต้น

คำสั่งที่ใช้ในที่ควบคุมเครื่องจักรจะแบ่งออกเป็นตระกูลตามความเหมาะสมในการใช้งาน เช่น G-code หรือ NC Program คือรหัสที่กำหนดตำแหน่งในแต่ละแนวแกนของเครื่องจักรโดยตรง ซึ่งผู้วิจัยได้เลือกชุดคำสั่งที่มีความซับซ้อนขึ้นนั่นก็คือ Robot programming language ซึ่งใช้ในการควบคุมแขนกลที่มีองศาอิสระ หรือ degrees of freedom (DOF) มากกว่า 6 แนวแกนขึ้นไป ในส่วนของ Positional Movement Instructions หรือการกำหนดจุดปลายทางของเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 รูปแบบการเคลื่อนที่แบบ PTP และ LIN

Positional Movement Instructions คือรหัสที่กำหนดการเคลื่อนไหวของแขนกลโดยกำหนดเพียงแค่จุดเริ่มต้นและจุดสิ้นสุดของการเคลื่อนไหว หน่วยประมวลผลภายในเครื่องจักรจะทำการประมวลผลการเคลื่อนไหวให้เหมาะสมกับจุดทั้ง 2 ตามรูปแบบที่กำหนดไว้ ชุดคำสั่งของบริษัท ABB และ KUKA ที่นิยมใช้นั้นได้แก่

- MoveJ / PTP - moves the TCP in a way which is easiest for the robot and not in a straight line
- MoveL / LIN moves the TCP in a way which is in a straight line
- MoveC/ CIRC complete circle.

Program	Configure	Monitor	Setup	Commands	Technology
PTP P1 CONT Ve1= 100 % PDAT14					
PTP P2 CONT Ve1= 100 % PDAT15					
\$TIMER_STOP[10]=FALSE					
\$TIMER[10]=0					
FOR I1=0 TO 10000					
PTP P3 Ve1= 100 % PDAT2					
OUT 1 : symName State= TRUE					
WAIT Time= 0.5 sec					
PTP P4 CONT Ve1= 100 % PDAT3					
PTP P5 CONT Ve1= 100 % PDAT4					
PTP P6 Ve1= 100 % PDAT5					
OUT 1 : symName State= FALSE					
WAIT Time= 0.5 sec					
P5 CONT Ve1=100 % PDAT6					
PTP P2 CONT Ve1= 100 % PDAT7					
ENDFOR					
RC:\R1\PROGRAM\DC Ln 24, Col 18					
no.	Source	Message			
01319	BDF	Office option enabled.			
01350		Programmed path reached (BC0)			
01350		Programmed path reached (BC0)			
01123		Approximation not possible			
CAPS S I R DOLP_R1 Line 20 TI HOV=50% R_Name					
Abort LIN/CIRC Touch Up Cm					

รูปที่ 6.3 ตัวอย่างโปรแกรมใน Robot programming language (KUKA version)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในคำสั่ง PTP และ MoveJ เป็นคำสั่งที่ไม่คำนึงถึงตำแหน่งระหว่างทาง แต่ระบบจะพยายามให้เป็นเส้นทางที่เรียบง่ายที่สุดสำหรับเครื่องจักรตัวนั้น ๆ [24] โดยมากจะถูกใช้ในกรณีที่ต้องการเวลาที่รวดเร็วและต้องการให้เครื่องจักรนั้นเคลื่อนไหวน้อยที่สุด เช่น การหยิบและวาง เป็นต้น การทำงานโดยทั่วไป จะกำหนดท่าทาง (Pose) โดยการปรับตำแหน่งของเครื่องจักรโดยใช้ pendant หรือ รีโมทคอนโทรล ของเครื่องจักร และกำหนดชนิดของคำสั่ง เป็นจุด (Point) จำนวนหลายจุดติดต่อกัน รวมทั้งเงื่อนไขการทำงานของโปรแกรม และจากนั้นจะทำการสั่งให้เครื่องจักรทำงานตามโปรแกรมที่ตั้งไว้ ลักษณะเดียวกับโปรแกรมที่ทำงานในคอมพิวเตอร์

เนื่องจากการเคลื่อนที่ในรูปแบบ PTP จะเป็นการเคลื่อนที่และหยุดตามจุดต่างๆ จนกระทั่งหมดโปรแกรม เป็นการทำงานอย่างเรียบง่าย แต่ในระบบอุตสาหกรรมในปัจจุบันนั้น เครื่องจักรหลาย ๆ ชนิดต้องการการคำนวณการเคลื่อนที่ที่ซับซ้อนยิ่งขึ้นเนื่องจากเหตุผลในหลาย ๆ ด้าน ได้แก่

- ความเร็วในการเคลื่อนที่ เพื่อเพิ่มความเร็วของกระบวนการการทำงาน
- แรงต้านการเคลื่อนที่ เพื่อลดความสึกหรอของตัวเครื่องจักร

ปัจจัยทั้ง 2 ชนิดนั้นสามารถทำให้ดีขึ้นได้โดยการคำนวณการเคลื่อนที่ไว้ล่วงหน้า ซึ่งต้องใช้ข้อมูลของเครื่องจักรและภาระ (Payload) ที่เครื่องจักรนั้น ติดตั้งอยู่

เพื่อประสิทธิภาพที่ดีขึ้นของการเคลื่อนไหวนั้นของเครื่องจักรทั้งทางด้านเวลาและทางด้านแรงต้าน ผู้วิจัยจึงทำการกำหนดสถานการณ์ดังกล่าวเป็นปัญหาทางการหาค่าที่เหมาะสมที่สุด เพื่อคำนวณความเร็วของการทำท่าทางที่ต่อเนื่องทั้งหมดในโปรแกรม ทำให้สามารถหาวิธีที่เหมาะสมที่สุดในด้านความเร็วและแรงต้าน ซึ่งเพื่อให้ปัญหานี้เหมาะสมกับการหาค่าที่เหมาะสมที่สุด ผู้วิจัยได้เลือกเครื่องจักรเป็นประเภทที่ต้องรับภาระสูงและภาระนั้นมีค่าความเฉื่อย เช่น แขนกลที่ต้องหยิบและวางสิ่งของที่มีน้ำหนักมาก มาเป็นปัญหาที่ประยุกต์อัลกอริทึม Pivot Optimizer

และทำการประยุกต์ระบบดังกล่าวลงในระบบสมองกลแบบฝังตัวเพื่อพิสูจน์ความยืดหยุ่นและความง่ายในการประยุกต์ลงในอุปกรณ์อื่นๆ นอกจากคอมพิวเตอร์

6.2 ปัญหา PTP motion optimization

จากเหตุผลที่ได้กล่าวไปในบทนำ ผู้วิจัยจึงจัดปัญหาดังกล่าวให้อยู่ในรูปของปัญหาเชิงการหาค่าที่เหมาะสมที่สุดโดยกำหนดให้ Payload หรือภาระที่ติดตั้งอยู่บนเครื่องจักร และความสามารถของอุปกรณ์ขับเคลื่อน (Actuator) หรือ มอเตอร์ มีผลต่อความเร่งสูงสุดของเครื่องจักร การคำนวณอย่างง่ายสามารถทำได้โดยคำนวณแรงเฉื่อยของภาระที่ติดตั้งอยู่บนตัวเครื่องจักรโดยไม่คำนึงถึงความเฉื่อยของเครื่องจักรตามสมการที่ (6.1)

$$J = mL^2 \quad (6.1)$$

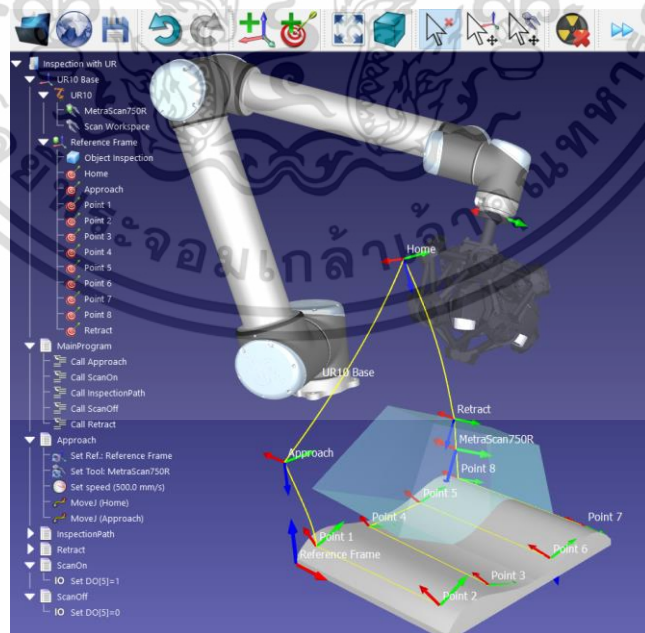
เมื่อ m คือมวลของภาระที่ติดตั้งมากับเครื่องจักร L คือระยะห่างจากมอเตอร์ไปยังจุดศูนย์กลางมวลของภาระ และ J คือความเฉื่อย (Inertia) ของภาระที่เกิดขึ้น เมื่อนำมาคำนวณร่วมกับสมการโมเมนต์ของแรง หรือ ทอร์ก (torque : τ) ซึ่งขึ้นอยู่กับความสามารถของมอเตอร์ ในสมการที่ (6.2)

$$\tau_a = JA \quad (6.2)$$

เมื่อ T_a คือ ค่าทอร์กเร่ง (Accelerating torque) ของมอเตอร์ J คือความเฉื่อยของระบบ และ A คือความเร่งของระบบ จะได้สมการที่อธิบายความสัมพันธ์ของมวลของภาระและความเร่งสูงสุดได้ตามสมการที่ 6.3 ซึ่งจะใช้เป็น เงื่อนไขความเร่งสูงสุดของการเคลื่อนที่ของเครื่องจักร

$$A = \frac{\tau_a}{mL^2} \quad (6.3)$$

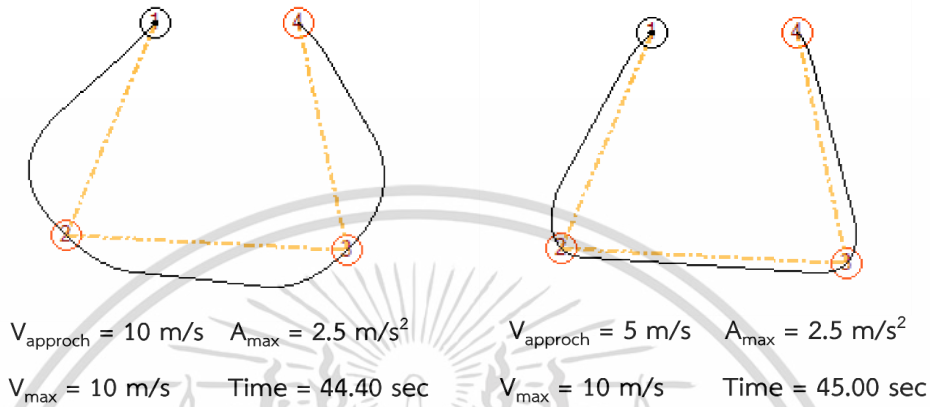
ตำแหน่งต่าง ๆ ที่เครื่องจักรจะต้องเคลื่อนที่ผ่าน จะถูกกำหนดไว้ก่อนการคำนวณ จะอยู่ในรูปของพิกัด ที่จำนวนมิติขึ้นอยู่กับ ความสามารถของเครื่องจักร เช่น เครื่องจักรที่มี 6 แกนอิสระ (6 DOF) สามารถกำหนดพิกัดได้ 6 ค่าใน 1 ตำแหน่งได้แก่ ตำแหน่งการเลื่อนในแกน (Translation) x y z และตำแหน่งเชิงหมุน (rotation) roll pitch yaw รวมเป็น 6 ค่าต่อ 1 พิกัด ดังรูปที่ 6.4



รูปที่ 6.4 การกำหนดพิกัดของเครื่องจักรที่มี 6 แกนอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

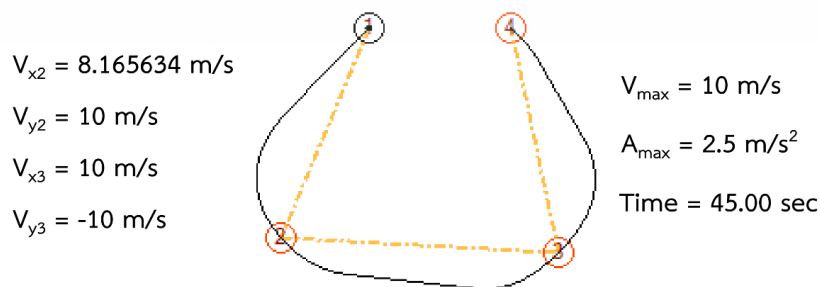
ซึ่งการเคลื่อนที่ตั้งแต่จุดเริ่มต้นไปยังจุดสุดท้ายโดยผ่านทุกจุดตามลำดับนั้น ทำได้หลายแบบมากเนื่องจากสามารถตั้งค่าความเร็วได้หลากหลายซึ่งค่าดังกล่าวเป็นจำนวนจริง จึงต้องใช้หลักการหาค่าเหมาะสมที่สุดในการหาค่าความเร็วที่เหมาะสมให้กระบวนการดังกล่าวจบลงด้วยเวลาที่เร็วที่สุด ตามตัวอย่างการเคลื่อนที่ในระนาบ 2 มิติ ตามรูปที่ 6.5



รูปที่ 6.5 เส้นทางที่เกิดเมื่อตั้งความเร็วเมื่อผ่านจุดให้มีค่าสูง (รูปซ้าย) และเส้นทางที่เกิดเมื่อตั้งความเร็วเมื่อผ่านจุดให้มีค่าต่ำ (รูปขวา)

จากรูปที่ 6.5 แสดงให้เห็นถึงรูปแบบการเคลื่อนที่ที่ต่างกันเมื่อปรับความเร็วในการเข้าถึงจุดที่ต่างกันออกไป โดยในรูปซ้ายมือจะแสดงการตั้งค่าความเร็วการเข้าสู่แต่ละจุดให้มีค่าเท่ากับความเร็วสูงสุดในทุกจุดทำให้ใช้เวลารวมไป 44.4 วินาที ซึ่งจะทำให้มีเวลาที่ดีกว่าการตั้งค่าความเร็วไว้ที่ครึ่งหนึ่งของความเร็วสูงสุด (รูปขวามือ) ซึ่งใช้เวลาไปทั้งหมด 45.0 วินาที แต่หากสังเกตจากวิธีการเคลื่อนที่ที่จะเห็นว่ารูปซ้ายมือนั้น เกิดการเคลื่อนที่ที่ สิ้นเปลืองมาก ซึ่งสามารถลดความสิ้นเปลืองนี้ทำให้เวลาดีขึ้นได้

ซึ่งหากใช้การหาค่าที่เหมาะสมที่สุดเข้าช่วยในการการหาความเร็วในการเข้าสู่จุดที่ทำให้การเคลื่อนที่ทั้งหมดใช้เวลาที่ดีที่สุดจะทำให้ได้การเคลื่อนที่ใหม่ที่ใช้เวลาเพียง 43.067 วินาที โดยค่าที่ได้จากการหาค่าที่เหมาะสมที่สุดจะเป็นดังรูปที่ 6.6



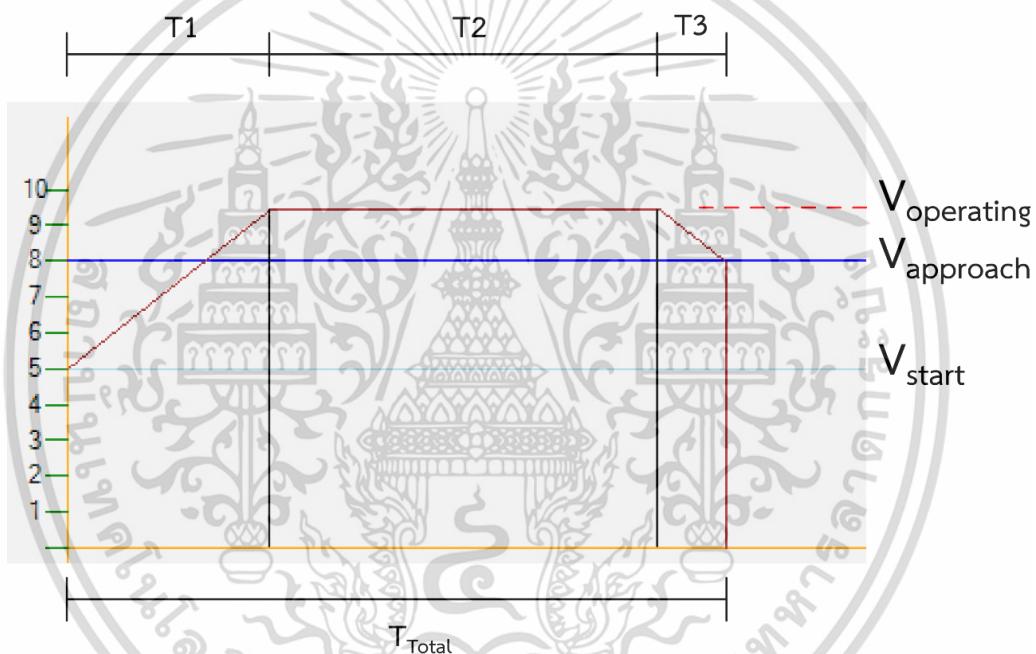
รูปที่ 6.6 เส้นทางเคลื่อนที่และการตั้งค่าที่ได้จากการหาค่าที่เหมาะสมที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.6 แสดงให้เห็นถึงผลลัพธ์ที่ดีที่สุดขึ้นหากใช้การหาค่าที่เหมาะสมที่สุดเข้ามาช่วย ซึ่งช่วยในเรื่องของการหาความเร็วที่จุดต่าง ๆ ทั้ง 2 แนวแกน โดยหากมีจำนวนจุดเป้าหมายมากขึ้นจะทำให้จำนวนของตัวแปร (Parameter) ที่ต้องการหาค่าที่เหมาะสมที่สุดนั้นมากขึ้น

การคำนวณหาเวลานั้น เนื่องจากจุดประสงค์ของระบบต้องการกำหนดความเร็วในการเข้าสู่จุดหมายของแต่ละแนวแกนแยกกันอย่างอิสระ แต่ต้องการให้เวลาที่เข้าถึงจุดหมายนั้น เข้าถึงพร้อมกันในทุกแนวแกน จึงจำเป็นต้องคำนวณหาการเคลื่อนที่ ที่เกิดขึ้นได้ทุกกรณีของแต่ละแนวแกน การเคลื่อนที่ของแต่ละแนวแกนนั้น จะแบ่งออกเป็น 3 ช่วง ได้แก่

- ช่วงการเร่งจากความเร็วต้นไปยังความเร็วที่ใช้งาน (T1)
- ช่วงการใช้ความเร็วคงที่ (T2)
- ช่วงการเร่งจากความเร็วใช้งานไปยังความเร็วปลาย (T3)

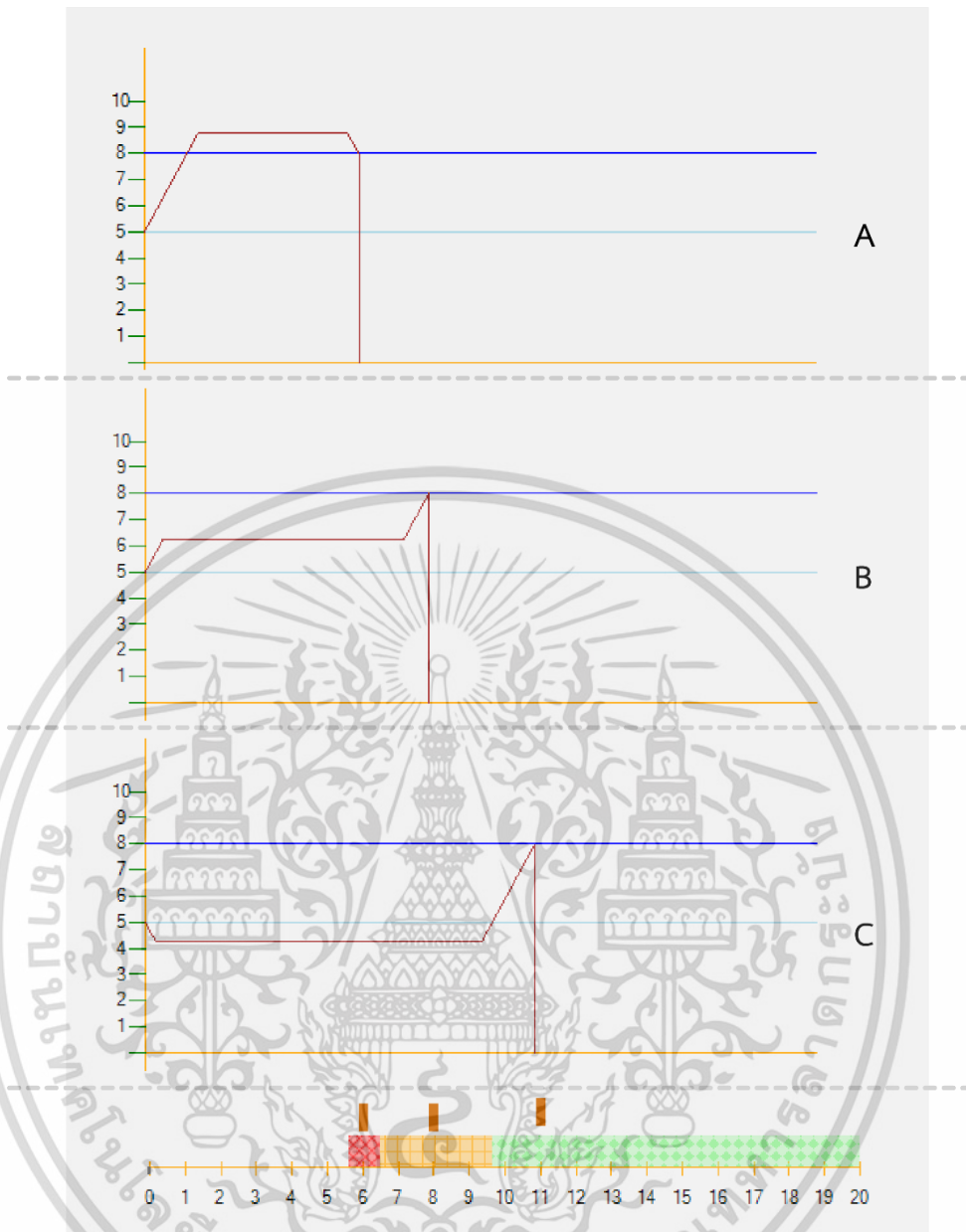


รูปที่ 6.7 ความเร็วตลอดการเคลื่อนที่จากจุดเริ่มต้นไปยังจุดเป้าหมาย

รูปที่ 6.7 อธิบายถึงความเร็วตลอดการเคลื่อนที่ที่สามารถเกิดขึ้นได้ ซึ่งเกิดจากการกำหนด V_{start} และ $V_{approach}$ จากการคำนวณจะทำให้เกิด รูปแบบของการเคลื่อนที่ออกมา ซึ่งรูปแบบการเคลื่อนที่ในรูปที่ 6.7 ดังกล่าวเป็นเพียงหนึ่งในกรณีที่สามารถเกิดขึ้นได้

การเคลื่อนที่ของเครื่องจักรในแกนหนึ่ง ๆ จากจุดเริ่มต้นไปยังจุดสุดท้ายคิดเป็นระยะทางหนึ่ง โดยการกำหนด ความเร็วเริ่มต้นจากจุดเริ่มต้น (V_{start}) ความเร็วสุดท้ายที่จุดเป้าหมาย ($V_{approach}$) จะทำให้เกิดการเคลื่อนที่ได้หลายรูปแบบ สิ่งที่เปลี่ยนแปลงไปในแต่ละกรณีนั้นคือ เวลาทั้งหมดที่ใช้ในการเคลื่อนที่ (T_{total}) ซึ่งเมื่อเปลี่ยนแปลงตัวแปร T_{total} จะทำให้ได้รูปแบบการเคลื่อนที่แบบอื่น ๆ ออกมาโดยที่ ตัวแปรที่ต้องการควบคุมซึ่งก็คือ V_{start} $V_{approach}$ และ ระยะทางรวม ยังมีค่าเท่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.8 กรณีของการเคลื่อนที่ต่าง ๆ ที่สามารถเกิดขึ้นได้

รูปที่ 6.8 แสดงตัวอย่างการเคลื่อนที่ แบบต่าง ๆ ตามการตั้งค่า เวลา (T_{total}) ที่ต่างกัน โดยแสดงการเคลื่อนที่ 3 แบบ ที่ใช้เวลา 6 8 และ 11 วินาที ตามลำดับ โดยหากจำแนกตาม ความเร็วใช้งาน ($V_{operate}$) ที่เกิดขึ้น จะสามารถแบ่งได้ 3 กรณีคือ

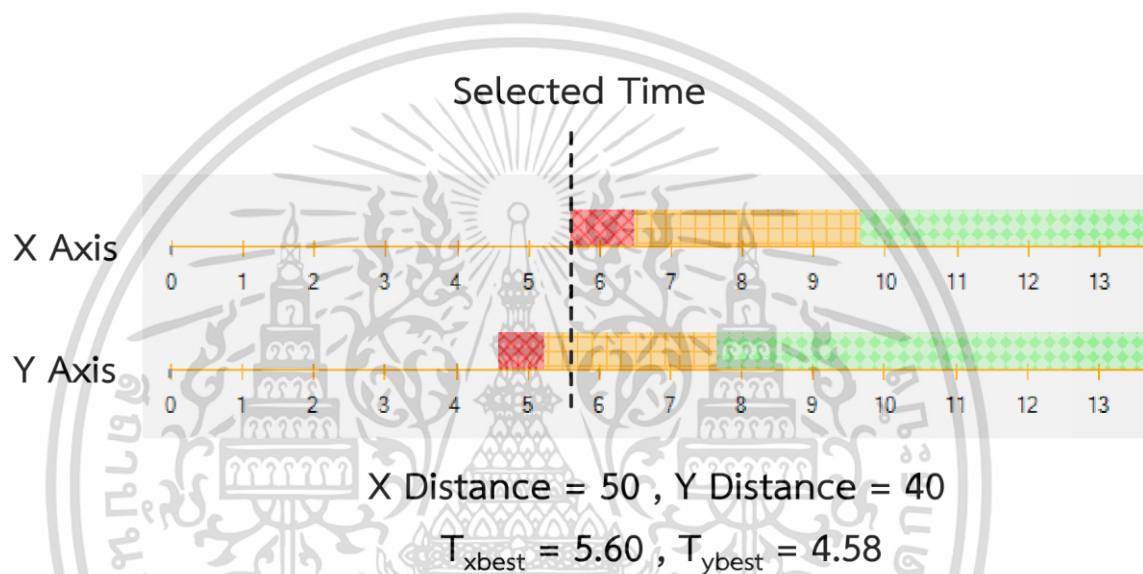
- กรณีแรก $V_{operate}$ มีค่ามากกว่า V_{start} และ $V_{approach}$ ($V_{operate} > V_{start}$ and $V_{operate} > V_{approach}$) กรณีจะทำให้เกิดการเคลื่อนที่ที่เร็วที่สุดหรือใช้เวลาน้อยที่สุดเกิดขึ้น
- กรณีที่สอง $V_{operate}$ มีค่าระหว่าง V_{start} และ $V_{approach}$ ($V_{approach} > V_{operate} > V_{start}$ or $V_{start} > V_{operate} > V_{approach}$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีที่สาม $V_{operate}$ มีค่าน้อยกว่า V_{start} และ $V_{approach}$ ($V_{operate} < V_{start}$ and $V_{operate} < V_{approach}$) จะทำให้เกิดการเคลื่อนที่ที่ใช้เวลานานที่สุด

โดยเมื่อใช้การเคลื่อนที่ทั้ง 3 กรณีมาประกอบกันจะทำให้ได้ความเป็นไปได้ทั้งหมดของการเคลื่อนที่ในแต่ละแนวแกนทำให้สามารถรู้เวลาที่เกิดขึ้นได้ของการเคลื่อนที่เริ่มต้นจากเวลาที่รวดเร็วที่สุด (T_{best}) ไปยังเวลาอนันต์ ดังที่แสดงในรูปที่ 6.8

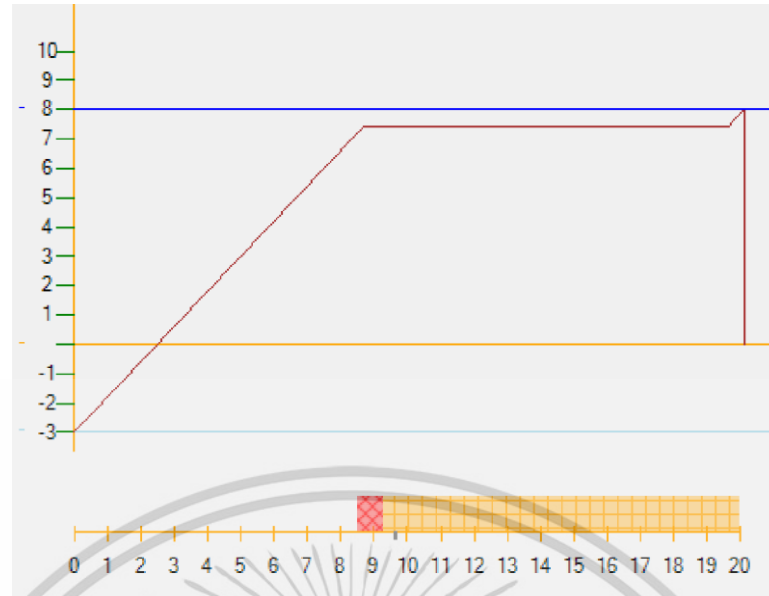
เวลาที่จะเลือกใช้ในการเคลื่อนที่นั้นจะนำเวลาที่ที่ดีที่สุดของทุกแนวแกน (ในที่นี้คือ 2 แนวแกน) การเคลื่อนที่มาพิจารณา เพื่อให้การเคลื่อนที่สามารทำได้จริงจึงต้องใช้เวลาที่ที่ดีที่สุดหรือเวลาที่เร็วที่สุดของ แกนการเคลื่อนที่ที่ใช้เวลานานที่สุดดังตัวอย่างต่อไปนี้



รูปที่ 6.9 ตัวอย่างการเลือกเวลาในการเคลื่อนที่ ระหว่างเป้าหมาย

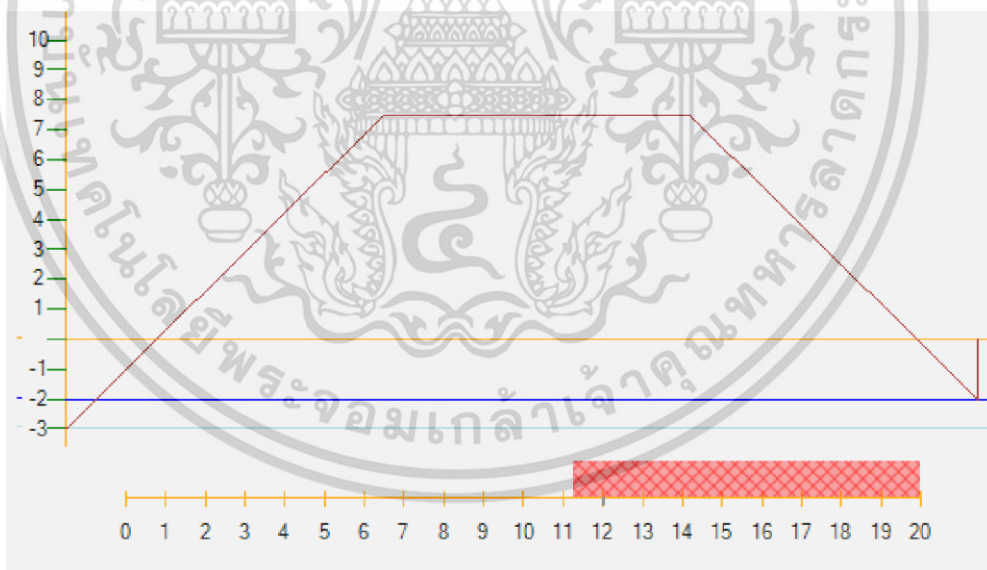
รูปที่ 6.9 แสดงถึงความเป็นไปได้ของการเคลื่อนที่ใน 2 แนวแกนซึ่งมีระยะทางการเคลื่อนที่ที่ต่างกัน ซึ่งในตัวอย่างมีระยะทาง 50 และ 40 หน่วย ตามลำดับซึ่งจะทำให้เกิดการเคลื่อนที่ที่ดีที่สุดโดยใช้เวลา 5.60 และ 4.58 ตามลำดับ เพื่อให้การเคลื่อนที่ที่เกิดขึ้นได้จริงการเคลื่อนที่ระหว่างจุดนี้ จะใช้เวลาเท่ากับ 4.58 วินาที ตามเส้น selected time ซึ่งแกน X จะเกิดการเคลื่อนที่ในกรณีแรก และแกน Y จะเกิดการเคลื่อนที่ในกรณีที่สอง

อย่างไรก็ตามในบางเงื่อนไขของการเคลื่อนที่จะทำให้การเคลื่อนที่ของระบบเกิดขึ้นไม่ครบทั้ง 3 กรณี แต่ก็ไม่ได้ส่งผลกระทบต่อความเป็นไปได้ที่เกิดขึ้นทั้งหมดเนื่องจากยังสามารถทำให้เกิดความเป็นไปได้ในการเคลื่อนที่จากเวลาที่ใช้นานน้อยที่สุดไปถึงเวลาอนันต์เสมอ



รูปที่ 6.10 การเคลื่อนที่ที่ไม่มีกรณีที่สาม

รูปที่ 6.10 คือกรณีที่ความเร็วต้นหรือความเร็วปลายอย่างใดอย่างหนึ่งมีค่าตรงกันข้ามกับทิศทางที่ต้องการเคลื่อนที่ ทำให้กรณีที่สองสามารถทำให้เกิดเวลาไปถึงเวลานั้นได้ จึงไม่จำเป็นต้องใช้กรณีที่สาม ซึ่งไม่สามารถเกิดขึ้นได้



รูปที่ 6.11 การเคลื่อนที่ที่มีเพียงกรณีแรก

รูปที่ 6.11 แสดงให้เห็นการเคลื่อนที่ที่ความเร็วต้นและความเร็วปลายมีทิศทางตรงกันข้ามกับทิศทางเคลื่อนที่ ทำให้การเคลื่อนที่ในกรณีแรกมีความกว้างไปจนถึงเวลานั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงเป็นที่มาของปัญหา motion optimization ซึ่งสามารถจัดให้อยู่ในรูปของปัญหาการหาค่าเหมาะสมที่สุดได้ดังสมการต่อไปนี้กำหนดให้ (Consider)

$$\begin{aligned} X &= [0, 0, x_1, x_2 \dots x_n, 0, 0] \\ &= [v_{x1}, v_{y1}, v_{x2}, v_{y2}, v_{x3}, v_{y3} \dots v_{xm}, v_{ym}] \end{aligned} \quad (6.4)$$

เมื่อ V_{ij} คือความเร็วปลายของการเข้าถึงเป้าหมายของการเคลื่อนไหวในแนวแกน i ในจุดที่ $j+1$ m คือจำนวนของตำแหน่งเป้าหมายทั้งหมดที่ต้องการให้เครื่องจักรเคลื่อนที่ผ่าน และ n คือจำนวนของตัวแปรที่ต้องการหาค่าที่เหมาะสมที่สุดซึ่งจะมีค่าเท่ากับ $(m * 2) - 2$ เสมอ โดยที่ความเร็วการเข้าถึงเป้าหมายของจุดแรกและจุดสุดท้ายจะมีค่าเท่ากับ 0 เนื่องจากการเริ่มต้นและสิ้นสุดการทำงาน

เพื่อให้ฟังก์ชันดังต่อไปนี้มีค่าน้อยที่สุด (Minimize)

$$F(X) = \sum_{i=1}^{m-1} \text{Max} \left(\begin{array}{l} \text{Time}(D_{xi}, v_{xi}, v_{xi+1}, a_{max}, v_{max}), \\ \text{Time}(D_{yi}, v_{yi}, v_{yi+1}, a_{max}, v_{max}) \end{array} \right) \quad (6.5)$$

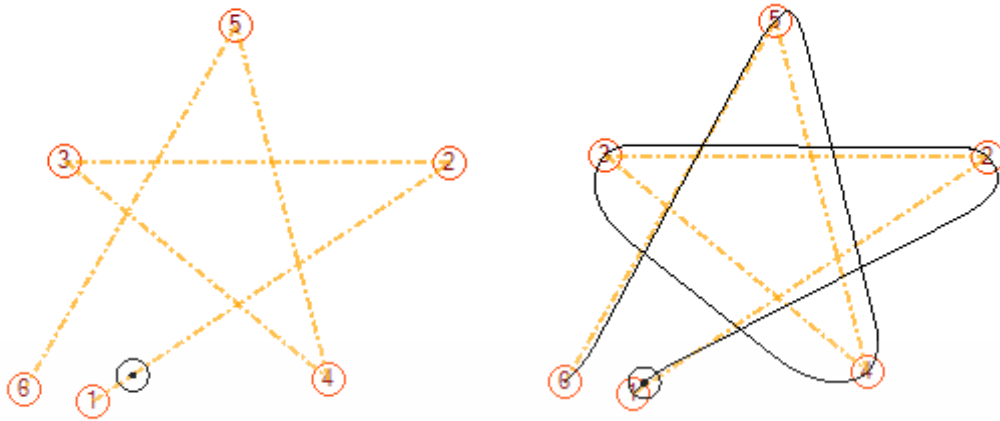
$$-v_{max} < v_x, v_y < v_{max} \quad (6.6)$$

เมื่อ $\text{Time}(D, U, V, A_{max}, V_{max})$ คือเวลาในการเคลื่อนที่ใช้เวลาน้อยที่สุด เป็นระยะทาง D โดยมีความเร็วเริ่มต้นเท่ากับ U และมีความเร็วปลายในการเข้าสู่ระยะทางเป้าหมายเป็น V ในตลอดทั้งการเคลื่อนที่นั้นจะใช้ความเร็วสูงสุดได้ไม่เกิน V_{max} และใช้ความเร่งสูงสุดได้ไม่เกิน A_{max} ซึ่งฟังก์ชันเป้าหมายดังกล่าวโดยจะทำการเลือกเวลาที่มากที่สุด เพื่อให้สอดคล้องกับทฤษฎีข้างต้น สุดท้ายจะได้เวลาทั้งหมดของการเคลื่อนที่ออกมาเพื่อทำการหาค่าที่ดีที่สุดต่อไป

เนื่องจากปัญหานี้กำหนด ช่วงของตัวแปรการตัดสินใจไว้ด้วย ขอบเขต boundary ซึ่งมีขอบเขตข้างมาก (upper bound) และ ขอบเขตข้างน้อย (lower bound) ไว้ด้วย ตัวแปร V_{max} จึงทำให้คำตอบที่ออกมาเป็นจริงได้ในทุกกรณีจึงถือว่าปัญหานี้ไม่มีข้อจำกัด (Unconstraint optimization problem) ทำให้คำตอบที่เกิดขึ้นในขอบเขตดังกล่าวนั้นใช้ได้เสมอ

ตัวอย่างเช่น การเคลื่อนที่ 6 จุดเป้าหมายโดยในแต่ละจุดเป้าหมายมีตำแหน่งดังต่อไปนี้ Target = {{208,218} , {385,100} , {194,99} , {325,207} , {279,32} , {174,212}} จะทำให้การเคลื่อนที่เป็นไปตามรูปที่ 6.12 ซึ่งจะทำให้ระยะห่างระหว่างจุดในแต่ละแนวการเคลื่อนที่เป็น Distance = {{177 , -118} , {-191,-1} , {131,108} , {-46,-175} , {-105,180}} ตามที่แสดงในรูปที่ 6.12 (ซ้าย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



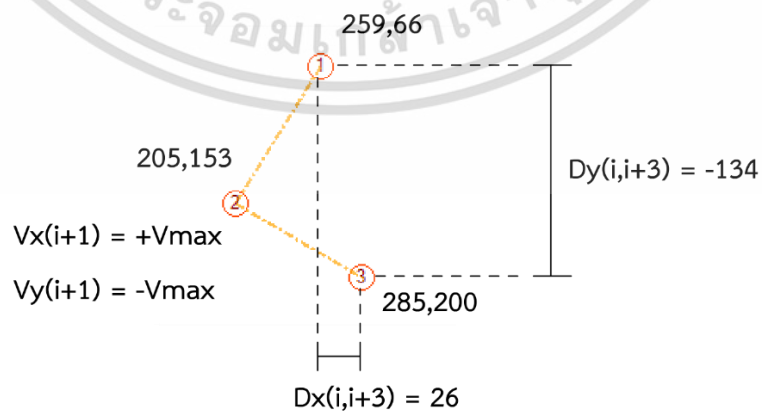
รูปที่ 6.12 ตัวอย่างคำถาม(รูปซ้าย) และคำตอบตั้งต้น(รูปขวา) ของเส้นทางการเคลื่อนที่

ในส่วนของการคำตอบตั้งต้น (Initial solution) ปัญหาการหาค่าเหมาะสมที่สุดนั้นส่วนใหญ่ มักจะสร้างคำตอบตั้งต้นด้วยการสุ่มตัวแปรการตัดสินใจทุกตัวเนื่องจากเป็นปัญหาเชิงคณิตศาสตร์ แต่ ปัญหา motion optimization นั้น ได้รับแรงบันดาลใจจากการเคลื่อนที่จริง จึงสามารถสร้าง คำตอบตั้งต้นที่มีแนวโน้มเป็นไปในทางที่ดี (convergence) ให้เกิดขึ้นได้

ความเร็วที่ใช้สร้างคำตอบตั้งต้นนั้น พิจารณาจากจุด สามจุดด้วยกันตามสมการที่ 6.7

$$V_{a(i+1)} = \begin{cases} +V_{max} & \text{if } D_a(i,i+3) > 0 \\ -V_{max} & \text{if } D_a(i,i+3) < 0 \\ 0 & \text{if } D_a(i,i+3) = 0 \end{cases} \quad (6.7)$$

กำหนดให้จุดที่ i , $i+1$ และ $i+2$ คือจุดเป้าหมายสามจุดที่ต่อเนื่องกัน และเรียงกันตามลำดับ และ a คือแกนที่ใช้ในการพิจารณา D_a คือระยะทางระหว่างแนวการเคลื่อนที่ a ระหว่างจุดสองจุด ความเร็วที่กำหนดของการเข้าหา ให้จะมีเครื่องหมายเท่ากับระยะห่างระหว่างสองจุดก่อนและหลัง



รูปที่ 6.13 การพิจารณาคำตอบตั้งต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.13 แสดงถึงตัวอย่างการคำนวณของจุดสามสุด ซึ่งจะแสดงค่าระยะห่างและค่าความเร็วในแนวแกน X และ Y โดยจะมีค่าเป็นค่าบวกและติดลบตามลำดับซึ่งจะสัมพันธ์กับสมการที่ (6.7) ซึ่งหากใช้หลักการคำตอบตั้งต้นดังกล่าวกับปัญหาข้างต้นในรูปที่ 6.12 (ซ้าย) จะทำให้ได้เส้นทางการเคลื่อนที่ตามรูปที่ 6.12 (ขวา)

6.3 การทดลองใช้การหาค่าที่เหมาะสมที่สุดเพื่อแก้ไขปัญหา

วิธีการหาเส้นทางการเคลื่อนที่ที่ได้กล่าวมาข้างต้นนั้นได้แก่ การเคลื่อนที่จุดต่อจุดอย่างง่าย (PTP/MoveJ) และ การเคลื่อนที่ระหว่างจุดอย่างต่อเนื่องโดยใช้คำตอบตั้งต้น จะให้ค่าเวลาเส้นทางที่คงที่ออกมาเสมอเนื่องจากเป็นวิธีการคำนวณโดยตรง (Direct method) ซึ่งสามารถนำการหาค่าที่เหมาะสมที่สุดเข้าช่วยได้โดยการหาค่าตัวแปรความเร็วเข้าสู่จุดเป้าหมายตามฟังก์ชันเป้าหมาย (Objective function) ตามสมการที่ (6.5) ตามเงื่อนไขความเร็วสูงสุดและความเร่งสูงสุดตามสมการที่ (6.6) และสมการที่ (6.7) ตามลำดับ

การทดลองจะใช้อัลกอริทึม ทั้งหมด 4 อัลกอริทึม เพื่อเปรียบเทียบประสิทธิภาพทางด้านผลลัพธ์นั้นคือเวลารวมในการเคลื่อนที่ของเครื่องจักรตามฟังก์ชันเป้าหมาย ได้แก่

- การเคลื่อนที่ระหว่างจุดอย่างง่าย (Point-to-point :PTP/MoveJ)
- คำตอบตั้งต้น (Initial solution : IS)
- การค้นหาเฉพาะที่ (Local search : LS)
- การค้นหาแบบแบ่งกลุ่มอนุภาค (Particle swarm optimization : PSO)
- การวิวัฒนาการโดยใช้ผลต่างแบบ (best/1/exp) (Differential Evolution : DE)
- การหาค่าที่เหมาะสมที่สุดแบบจุดหมุน (Pivot optimizer : PO)

PTP นั้นใช้เพื่อการอ้างอิงประสิทธิภาพพื้นฐานที่ระบบทั่วไปจะสามารถคำนวณได้และเป็นรูปแบบทั่วไปที่เครื่องจักรประเภทแขนกลใช้ในการทำงาน เนื่องจากปัญหานี้ไม่ได้สร้างคำตอบตั้งต้นโดยการสุ่ม IS จึงสามารถถูกใช้เปรียบเทียบเพื่อประสิทธิภาพที่เพิ่มขึ้นจากอัลกอริทึมต่าง ๆ ที่นำมาประยุกต์ LS นั้นเป็นอัลกอริทึมที่หาได้เฉพาะจุดต่ำสุดสัมผัสจากคำตอบตั้งต้น ถูกนำมาอ้างอิงประสิทธิภาพการหาค่าตอบของอัลกอริทึมอื่นในด้านของการหาค่าตอบในปัญหาแบบพหุฐานนิยม (Multimodal problem) และเป็นตัวช่วยยืนยันความซับซ้อนของปัญหา

PSO DE และ PO คืออัลกอริทึมการหาค่าที่เหมาะสมที่สุดเพื่อวัดประสิทธิภาพโดยจะทำการกำหนดจำนวนรอบไว้ที่ 1000 เท่าของมิติของปัญหาและทำการประมวลผลทั้งหมด 25 ครั้งเพื่อนำผลที่ดีที่สุดมาบันทึกและอภิปรายผลต่อไป

เนื่องจาก DE นั้นใช้แนวความคิดหลักจาก วิธีเชิงพันธุกรรม (genetic algorithm) จึงทำให้ไม่สามารถใช้ IS มาเป็นคำตอบตั้งต้นเพียงอย่างเดียวได้จึงจำเป็นต้องสร้างคำตอบตั้งต้นที่เกิดจากการสุ่มขึ้นมา ซึ่งในที่นี้จะสุ่มค่าจากคำตอบตั้งต้นที่ใช้ความเร็วครึ่งหนึ่ง และความเร็วสูงสุดแต่ทิศทาง

ตรงกันข้าม มาสู่ผสมลงในตัวแปรการตัดสินใจ เพื่อให้อัลกอริทึมมีประสิทธิภาพ และตั้งค่าความแปรปรวนเป็น $best/1/exp$ เพื่อให้ใช้ค่าตอบตั้งต้นเป็นหลัก

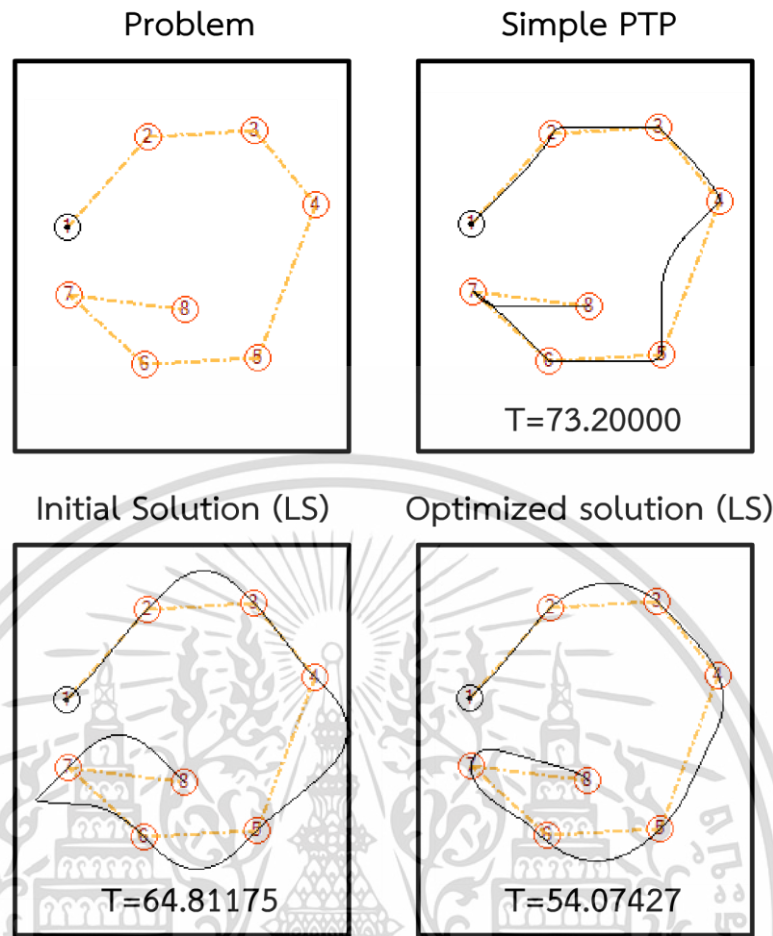
โจทย์ปัญหาที่ใช้นั้นจะมีทั้งหมด 10 โจทย์ปัญหา ตาม ภาคผนวก ข. โจทย์ปัญหาการเคลื่อนที่ในการทดลอง (Motion Problem Set) ซึ่งจะมีจำนวนของจุดเป้าหมายและตำแหน่งที่แตกต่างกันไป แต่เนื่องจากปัญหานี้ไม่สามารถหาคำตอบที่แท้จริงได้ (Global optimum) จึงต้องใช้เวลาซึ่งเป็นผลลัพธ์ในการเปรียบเทียบ จากเดิมที่ใช้คำตอบของโจทย์ปัญหา

โจทย์ปัญหาทั้ง 10 ปัญหาเรียงจากมิติน้อยไปยังมิตินอก หมายความว่า ความซับซ้อนจะเพิ่มขึ้นจากน้อยไปมาก ซึ่งจุดเป้าหมายในโจทย์ปัญหาที่ใช้จะมีดังนี้ 4 5 7 8 8 17 17 17 20 และ 20 ตามลำดับ ซึ่งมีมิติของปัญหาจะมีค่าเป็นสองเท่าของจุดเป้าหมายตามที่ได้อธิบายตามข้างต้น

6.4 ผลการทดลอง

ผลการทดลองในหัวข้อที่ 6.3 จะถูกบันทึกลงในตาราง ที่ 6.1 ลงในรูปของเวลารวมที่ดีที่สุดของการทำงานทวนรอบในแต่ละอัลกอริทึมตัวอย่างเช่น ปัญหาที่ 3 มีทั้งหมด 8 จุดเป้าหมายด้วยกันดังที่แสดงในรูป 6.14 (บนซ้าย) ซึ่งเป็นปัญหาที่มี 6 มิติการค้นหา การเคลื่อนที่ทั่วไปจะใช้หลักการของ simple PTP หรือการกำหนดให้ความเร็วต้นและความเร็วปลายมีค่าเท่ากับศูนย์ดังที่กล่าวไปข้างต้น (Stop-go PTP motion) ดังรูปที่ 6.14 (บนขวา) จะได้ความเร็วที่เป็นมาตรฐานสามารถใช้ในการเปรียบเทียบประสิทธิภาพที่เพิ่มขึ้นของอัลกอริทึม

การทดลองจะเริ่มจากการหาคำตอบตั้งต้นหรือ LS ดังรูปที่ 6.14 (ล่างซ้าย) ตามวิธีที่ได้กล่าวไปข้างต้น และใช้อัลกอริทึมทั้ง 3 ประยุกต์เข้ากับคำตอบตั้งต้นดังกล่าวเพื่อให้ได้ผลที่ดีขึ้น เช่น การใช้ LS เพื่อพัฒนาผลลัพธ์ที่ดีขึ้นในรูปที่ 6.14 (ล่างขวา) และบันทึกผลการทดลองลงในตาราง



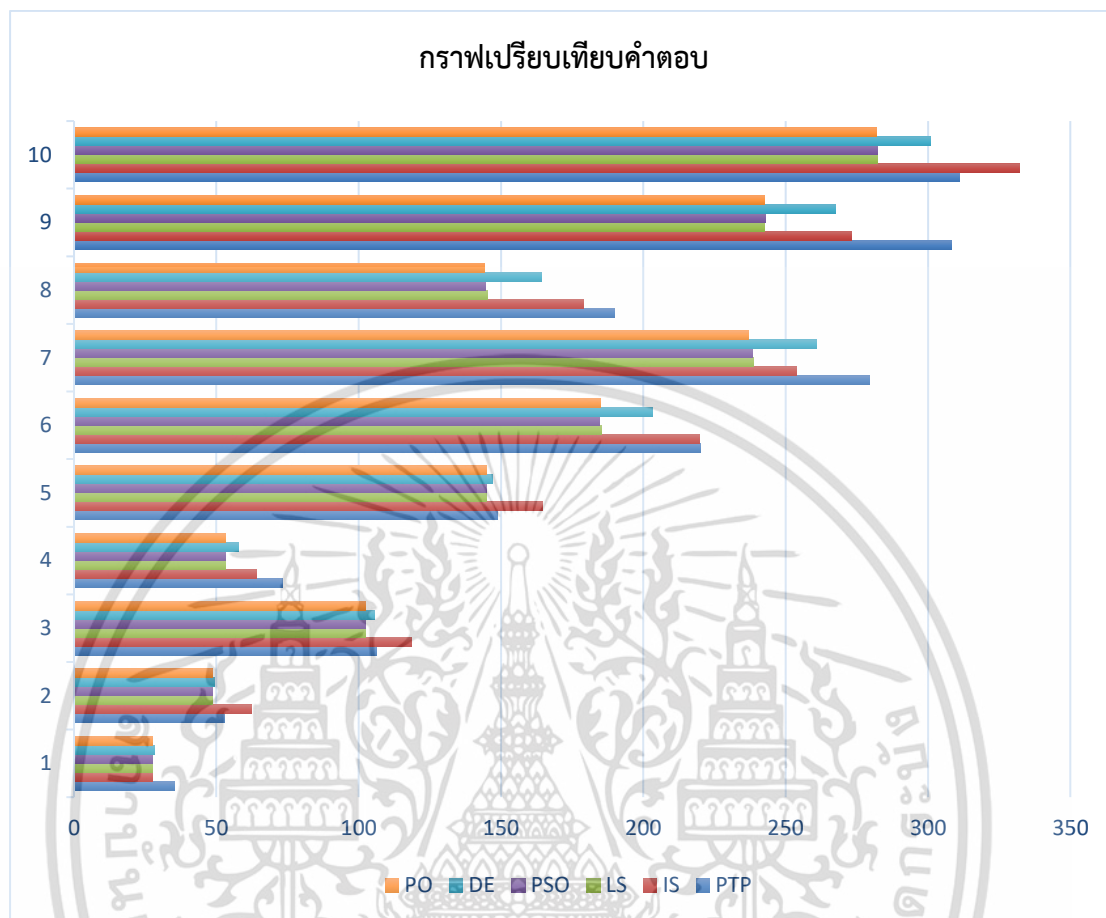
รูปที่ 6.14 ตัวอย่างการหาคำตอบของปัญหาการเคลื่อนที่

ตารางที่ 6.1 ผลการทดลองปัญหาการเคลื่อนที่

No.	PTP	IS	LS	PSO	DE	PO
1	35.40000	27.70030	27.40317	27.40318	28.15717	27.40319
2	52.70000	62.50040	48.70041	48.70040	49.44400	48.70042
3	106.40000	118.40060	102.40221	102.40060	105.50166	102.40252
4	73.20000	64.11245	53.36922	53.35778	57.64384	53.36036
5	148.60000	164.60070	144.91626	144.89984	147.14460	144.97721
6	220.20000	219.90160	185.21130	184.55346	203.14291	185.07181
7	279.37278	253.90160	238.60051	238.43839	260.79506	237.01347
8	190.00000	179.00000	145.12496	144.48255	164.11828	144.31940
9	308.17747	273.10190	242.53506	242.82013	267.39771	242.65016
10	311.00000	332.00190	282.36067	282.09923	300.80867	282.07962

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 สรุปผลการทดลอง



รูปที่ 6.15 กราฟเปรียบเทียบคำตอบของแต่ละวิธีการ

จากค่าเวลาผลลัพธ์ในตารางที่ 6.1 แสดงให้เห็นถึงเวลาที่เกิดจากกระบวนการต่าง ๆ จะเห็นได้ว่า การเคลื่อนที่แบบไปและหยุดหรือ PTP แบบธรรมดานั้นจะใช้เวลานานที่สุดในหลายๆ กรณี นอกจากโจทย์ปัญหาที่จุดเป้าเกิดการเปลี่ยนแปลงทิศทางเป็นส่วนมาก ซึ่งจะทำให้การหาคำตอบตั้งต้นอย่างง่ายนั้น หาคำตอบได้แย่มาก ส่วนของคำตอบตั้งต้นหรือ IS นั้นในกรณีทั่วไปจะทำให้เวลาลดลงจาก PTP มาก แต่เนื่องจากในบางกรณีการใช้ความเร็วสูงสุดนั้นไม่เป็นผลดีต่อการเคลื่อนที่ของเครื่องจักร จึงทำให้ได้คำตอบที่แย่ออกมา

อัลกอริทึม local search (LS) ที่ใช้ IS เป็นคำตอบตั้งต้นนั้น ให้ผลที่น่าพอใจอย่างมาก เนื่องจากได้ผลดีกว่า PTP และ IS ในทุกโจทย์ปัญหาแต่เนื่องจากการใช้คำตอบตั้งต้นที่ไม่ได้เกิดจากการสุ่มนั้นทำให้คุณสมบัติของความแปรปรวนหายไปจึงทำให้ได้คำตอบในสัมพัทธ์ที่ใกล้เคียงเดิมทุกครั้ง PSO ให้ผลที่ดีกว่า LS ในทุกปัญหาเนื่องจากสามารถข้ามสัมพัทธ์และมีความแปรปรวนที่เกิดขึ้นจากการใช้ประชากรทำให้ ได้ผลลัพธ์ที่หลากหลายและดีกว่า LS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DE นั้นเนื่องจากเป็นอัลกอริทึมที่ไม่สามารถทำให้คำตอบไปถึงจุดต่ำสุดสัมพัทธ์ได้ จึงให้ผลที่ไม่ดีนักเมื่อเทียบกับ PSO และ LS การสุ่มคำตอบตั้งต้นนั้นช่วยในเรื่องของการหาคำตอบที่แปรปรวนได้มากขึ้นแต่ยังไม่สามารถทำให้คำตอบนั้นออกมาดีใกล้เคียงกับ PSO และ LS ได้

Pivot Optimizer นั้น ในปัญหาข้อที่ 1 ถึง 5 ให้คำตอบที่ใกล้เคียงกับ LS และ PSO แต่แยกว่าทั้งสองอัลกอริทึมเนื่องจาก ปัญหานี้มีสมพัทธ์ที่ค่อนข้างกว้างและห่างกันมากจึงเสียความสามารถไปกับการหาสมพัทธ์จนไม่สามารถหาคำตอบที่ดีที่สุดได้ แต่ในปัญหาที่ซับซ้อนมากขึ้น การหาสมพัทธ์ที่หลากหลายขึ้นเป็นผลให้อัลกอริทึม ได้ผลลัพธ์ที่ดีมากขึ้นโดยมีดีกว่า PSO และ LS เป็นส่วนใหญ่ จะเห็นได้จาก กราฟเปรียบเทียบคำตอบที่ดีที่สุดของแต่ละอัลกอริทึมในรูปที่ 6.15



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Rao, Singiresu S. Engineering optimization: theory and practice. John Wiley & Sons, 2019.
- [2] Arora, Rajesh Kumar. Optimization: algorithms and applications. CRC Press, 2015.
- [3] Simon, Dan. Evolutionary optimization algorithms. John Wiley & Sons, 2013.
- [4] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-international conference on neural networks. Vol. 4. IEEE, 1995.
- [5] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." Advances in engineering software 69 (2014): 46-61.
- [6] Storn, Rainer, and Kenneth Price. "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces." Journal of global optimization 11.4 (1997): 341-359.
- [7] Elsayed, Saber M., Ruhul A. Sarker, and Daryl L. Essam. "Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems." 2011 IEEE Congress of Evolutionary Computation (CEC). IEEE, 2011.
- [8] Karaboga, Dervis, and Bahriye Basturk. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." Journal of global optimization 39.3 (2007): 459-471.
- [9] Liang, Jing J., Bo Y. Qu, and Ponnuthurai N. Suganthan. "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization." Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 635 (2013): 490.
- [10] Liang, J. J., et al. "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization." Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 201212.34 (2013): 281-295.
- [11] Suganthan, Ponnuthurai N., et al. "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization." KanGAL report 2005005.2005 (2005): 2005.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [12] S. Lensgraf and R. R. Mettu, "Incorporating Kinematic Properties into Fused Deposition Toolpath Optimization," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 8622-8627, doi: 10.1109/IROS.2018.8594398.
- [13] Q. Zhang and S. Li, "Minimum time feed-rate optimization along predefined toolpath with acceleration constraints on each axis," Proceedings of the 30th Chinese Control Conference, Yantai, China, 2011, pp. 5462-5467.
- [14] K. Erwinski, M. Paprocki, A. Wawrzak and L. M. Grzesiak, "PSO based feedrate optimization with contour error constraints for NURBS toolpaths," 2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 2016, pp. 1200-1205, doi: 10.1109/MMAR.2016.7575309.
- [15] S. Thongkrait and V. Chutchavong, "A Time Improvement PSO Base Algorithm Using Multithread Programming," 2019 4th International Conference on Communication and Information Systems (ICCIS), Wuhan, China, 2019, pp. 212-216, doi: 10.1109/ICCIS49662.2019.00044.
- [16] S. Thongkrait and V. Chutchavong, "Hard Deterministic Particle Swarm Optimisation for Certain Result Solution" International Conference on Electronics, Information, and Communication 2021. IEEE, 2021, pp. 629-632
- [17] Reinelt, Gerhard. "TSPLIB—A traveling salesman problem library." ORSA journal on computing 3.4 (1991): 376-384.
- [18] Rokbani, Nizar, Aijth Abraham, and Adel M. Alimi. "Fuzzy ant supervised by PSO and simplified ant supervised PSO applied to TSP." 13th International Conference on Hybrid Intelligent Systems (HIS 2013). IEEE, 2013.
- [19] J. Du, F. Zhang, G. Huang and J. Yang, "A new initializing mechanism in Particle Swarm Optimization," 2011 IEEE International Conference on Computer Science and Automation Engineering, Shanghai, 2011, pp. 325-329, doi: 10.1109/CSAE.2011.5952861.
- [20] Biscani, Francesco, and Dario Izzo. "esa/pagmo2: pagmo 2.15." zndo (2020), doi:10.5281/zenodo.1045336
- [21] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." Advances in engineering software 69 (2014): 46-61.

[22] Storn, Rainer, and Kenneth Price. "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces." *Journal of global optimization* 11.4 (1997): 341-359.

[23] Somsin Thongkrait, Vanvisa Chutchavong. "A Novel Evolution Optimization Algorithm Using a Multidimensional Geometric Method: Pivot Optimiser" *International Journal of Simulation- Systems, Science and Technology* N4 (2020), doi: 10.5013/IJSSST.a.21.04.13

[24] Enien, Karim, Tim Wrütz, and Rolf Biesenbach. "Development of a PTP Movement of a Industrial Robot via MATLAB by deriving its Kinematics and Integration in an Offline Programming Tool." 2018 19th International Conference on Research and Education in Mechatronics (REM). IEEE, 2018.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ฟังก์ชันทดสอบที่ใช้ในการทดลอง (Test Function)

รายการฟังก์ชันทดสอบทั้งหมดจะมีทั้งหมด 20 ฟังก์ชัน ตามที่ได้กล่าวไปในการทดลอง 6 การทดลอง ซึ่งจะแบ่งเป็นฟังก์ชัน 4 ประเภท ได้แก่

- ฟังก์ชันมาตรฐานในการประเมินผลอัลกอริทึมการหาค่าเหมาะสมที่สุด (F1 ถึง F6)
- การหมุนและเพิ่มระดับฟังก์ชันมาตรฐาน (Shifted and Rotated) (F7 ถึง F11)
- ฟังก์ชันที่ใช้ใน CEC (F12 ถึง F16)
- ฟังก์ชันผสม (Hybrid Function) ของ ฟังก์ชัน CEC (F17 ถึง F20)

โดยในแต่ละฟังก์ชันจะมีและข้อกำหนดตัวแปรเพื่อทำให้รูปแบบของฟังก์ชันเกิดการเปลี่ยนแปลงไปจากฟังก์ชันต้นฉบับซึ่งจะทำให้หาค่าตอบที่ดีที่สุดได้ยากขึ้นตามกฎของ CEC14 ดังนี้

- ฟังก์ชันทุกฟังก์ชันจะเป็นปัญหาการหาค่าที่ต่ำที่สุด (Minimize problem) โดยจะกำหนดให้ D คือมิติของปัญหา ค่าตอบที่ออกมาจะต้องอยู่ในรูปของฟังก์ชันการตัดสินใจ (decision vector) ในรอบที่ T ในรูปของ

$$\min f(x), x = [x_1, x_2, \dots, x_D]^T$$

- มีตัวแปร o ในการยกระดับฟังก์ชัน (Shifted global optimum) ซึ่งเป็นค่าคงที่แบ่งเป็นหลายชุดมีค่าในช่วง $[-80, 80]$ จำนวน D มิติ เกิดจากการสุ่มแบบอิสระต่อกันอยู่ในรูปของ ซึ่ง i คือหมายเลขของชุด

$$o_i = [o_{i1}, o_{i2}, \dots, o_{iD}]^T$$

- ปริภูมิการค้นหา (Search Space) คือจำนวนจริงในช่วง $[-100, 100]$ ในทุกมิติ
- มีตัวแปร M คือเมทริกซ์การหมุน (Rotation Matrix) ใช้ในการหมุนตัวแปรการตัดสินใจ เป็นค่าคงที่ที่เกิดจากการสุ่มแบบอิสระ

ฟังก์ชันในการทดสอบนั้นจะประกอบขึ้นจากฟังก์ชันพื้นฐานซึ่งจะใช้ตัวอักษร f พิมพ์เล็กดังนี้

f1. High Conditioned Elliptic Function

$$f_1(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2 \quad (\text{ก.1})$$

f2. Bent Cigar Function

$$f_2(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2 \quad (\text{ก.2})$$

f3. Discus Function

$$f_3(x) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2 \quad (\text{ก.3})$$

f4. Rosenbrock's Function

$$f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \quad (\text{ก.4})$$

f5. Ackley's Function

$$f_5(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e \quad (\text{ก.5})$$

f6. Weierstrass Function

$$f_6(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k \cdot 0.5)] \quad (\text{ก.6})$$

$$a = 0.5, b = 3, k_{max} = 20$$

f7. Griewank's Function

$$F_7(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (\text{ก.7})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f8. Rastrigin's Function

$$f_8(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (\text{ก.8})$$

f9. Modified Schwefel's Function

$$f_9(x) = 418.9829 \times D - \sum_{i=1}^D g(z_i) \quad (\text{ก.9})$$

$$z_i = x_i + 5.209687462275036e + 002$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{\frac{1}{2}}) & : \text{if } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|500 - \text{mod}(z_i, 500)|}) - \frac{(z_i - 500)^2}{1000D} & : \text{if } z_i > 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|\text{mod}(|z_i|, 500) - 500|}) - \frac{(z_i - 500)^2}{1000D} & : \text{if } z_i < -500 \end{cases}$$

f10. Katsuura Function

$$f_{10}(x) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{|2^j x_i - \text{round}(2^j x_i)|}{2^j} \right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} \quad (\text{ก.10})$$

f11. HappyCat Function

$$f_{11}(x) = \left| \sum_{i=1}^D x_i^2 - D \right|^{1/4} + (0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i) / D + 0.5 \quad (\text{ก.11})$$

f12. HGBat Function

$$f_{12}(x) = \left| \left(\sum_{i=1}^D x_i^2 \right)^2 - \left(\sum_{i=1}^D x_i \right)^2 \right|^{1/2} + (0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i) / D + 0.5 \quad (\text{ก.12})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f13. Expanded Griewank's plus Rosenbrock's Function

$$f_{13}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \dots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1)) \quad (\text{ก.13})$$

f14. Expanded Scaffer's F6 Function

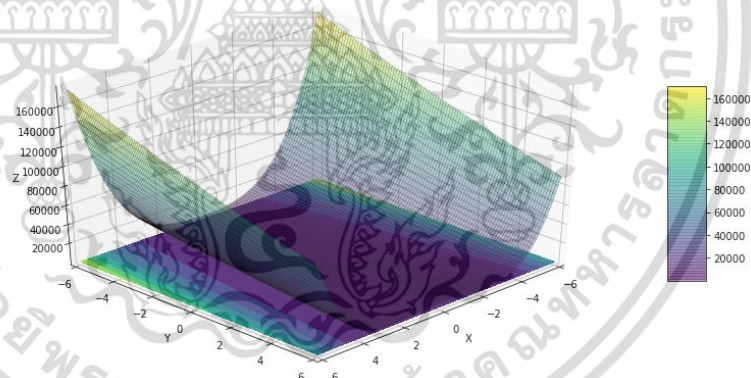
$$\text{Scaffer's F6 Function: } g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}$$

$$f_{14}(x) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1) \quad (\text{ก.14})$$

ซึ่งฟังก์ชันพื้นฐานเหล่านี้จะนำไปประกอบเป็นฟังก์ชันในการทดสอบทั้ง 20 ฟังก์ชันแทนด้วยตัวอักษร F พิมพ์ใหญ่ดังต่อไปนี้

1. Rosenbrock function (F1)

$$F_1(x) = f_4(x)$$

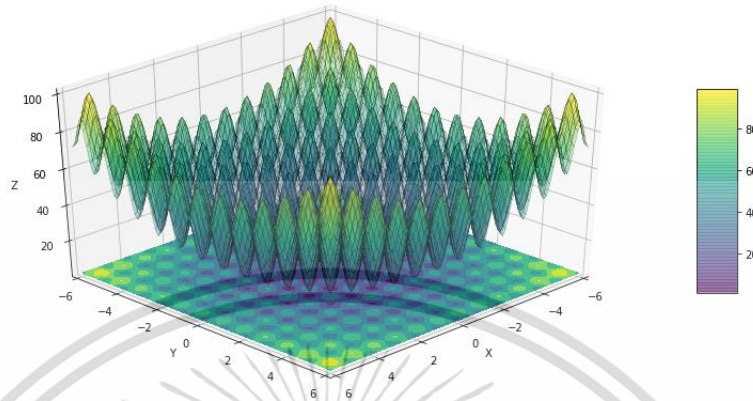


รูปที่ ก1.1 ฟังก์ชัน Rosenbrock ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Rastrigin function (F2)

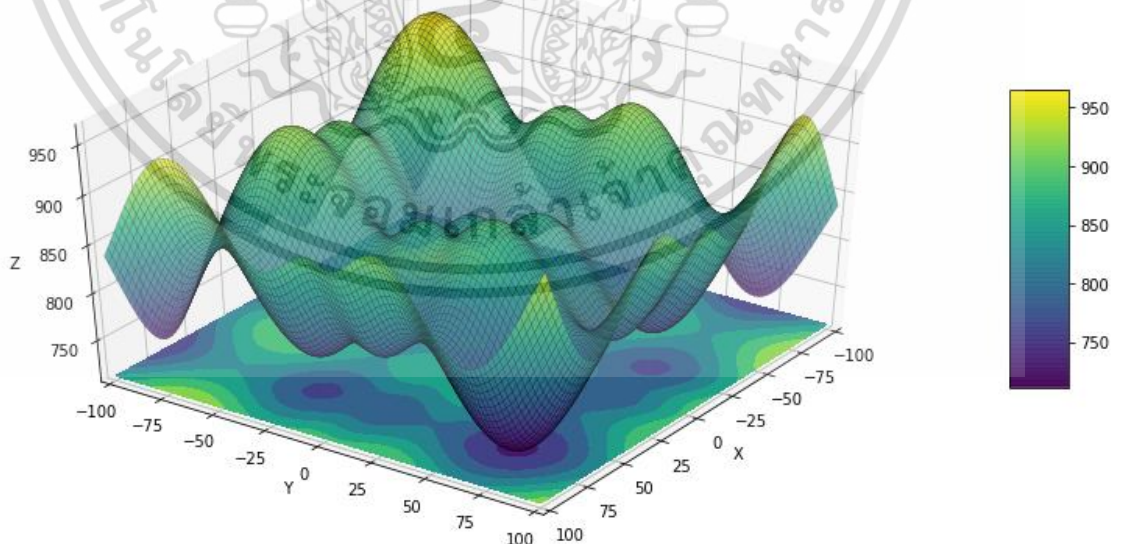
$$F_2(x) = f_8(x)$$



รูปที่ ก1.2 ฟังก์ชัน Rastrigin ในระนาบ 2 มิติ

3. Schwefel function (F3)

$$F_3(x) = 418.9829d - \sum_{i=1}^d x_i \sin \sqrt{|x_i|}$$

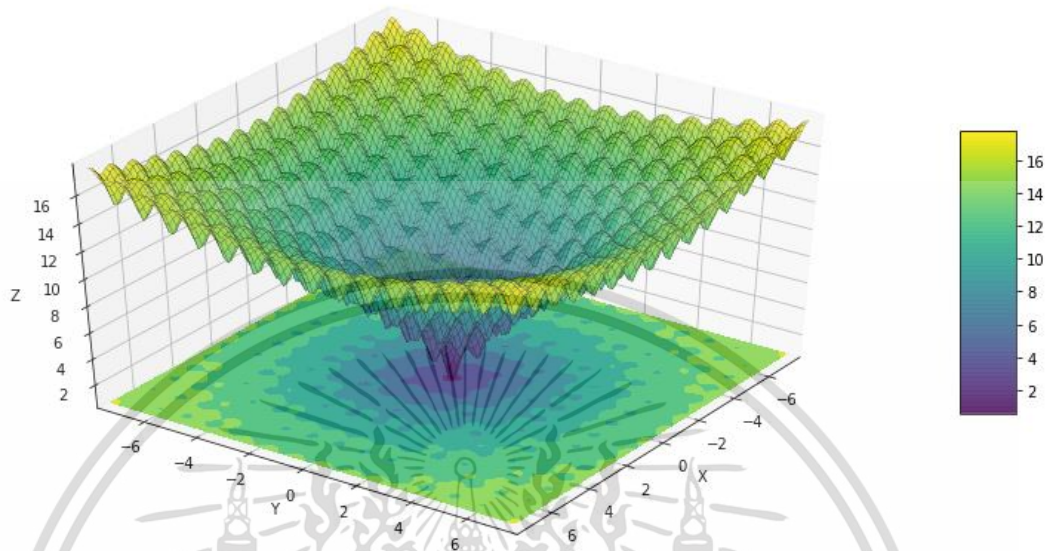


รูปที่ ก1.3 ฟังก์ชัน Schwefel ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Ackley function (F4)

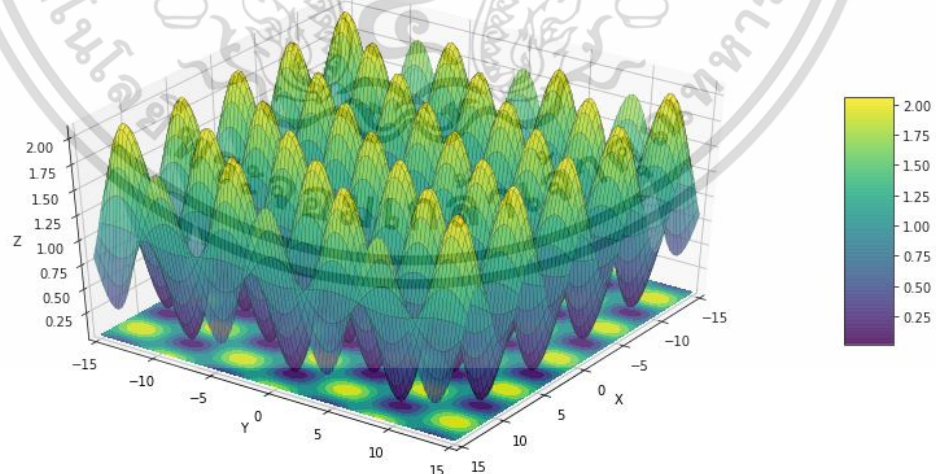
$$F_4(x) = f_5(x)$$



รูปที่ ก1.4 ฟังก์ชัน Ackley ในระนาบ 2 มิติ

5. Griewank function (F5)

$$F_5(x) = f_7(x)$$

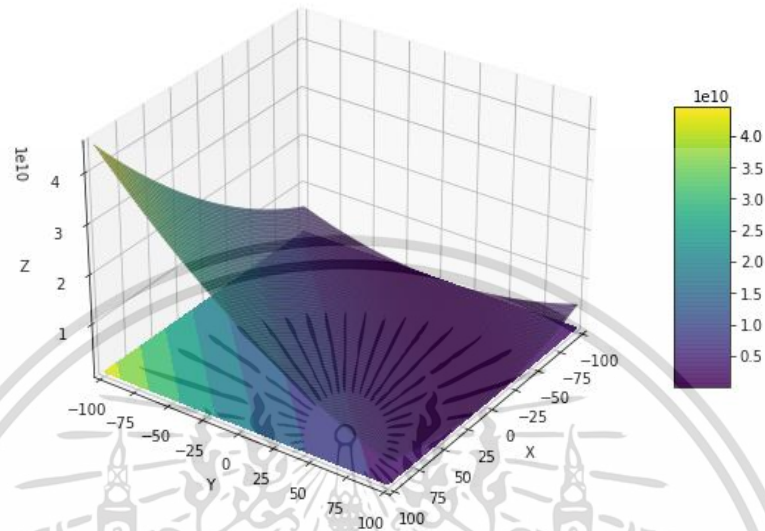


รูปที่ ก1.5 ฟังก์ชัน Griewank ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Rotated Bent Cigar Function (F6) (CEC2014 no.2)

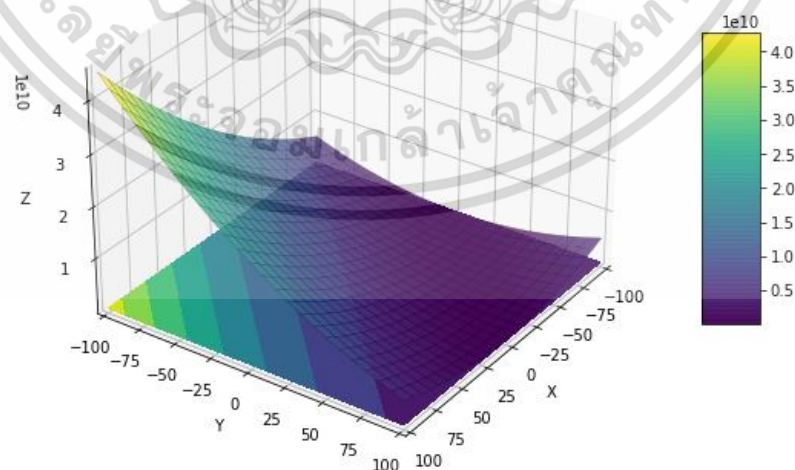
$$F_6(x) = f_2(M(x - o_2)) + 200$$



รูปที่ ก1.6 ฟังก์ชัน Rotated Bent Cigar ในระนาบ 2 มิติ

7. Rotated Discus Function (F7) (CEC2014 no.3)

$$F_7(x) = f_2(M(x - o_2)) + 300$$

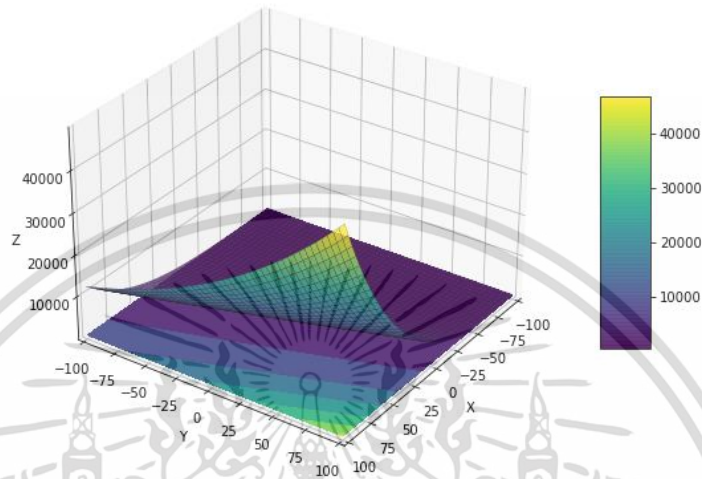


รูปที่ ก1.7 ฟังก์ชัน Rotated Discus Function ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. Shifted and Rotated Rosenbrock's Function (F8) (CEC2014 no.4)

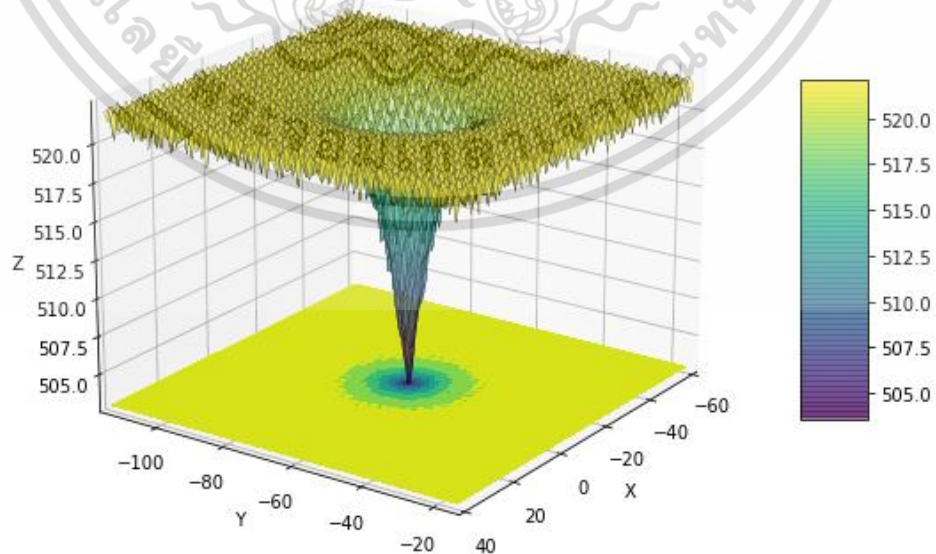
$$F_8 = f_4 \left(M \left(\frac{2.048(x - o_4)}{100} \right) + 1 \right) + 400$$



รูปที่ ก1.8 ฟังก์ชัน Shifted and Rotated Rosenbrock ในระนาบ 2 มิติ

9. Shifted and Rotated Ackley's Function (F9) (CEC2014 no.5)

$$F_9 = f_5(M(x - o_5)) + 500$$

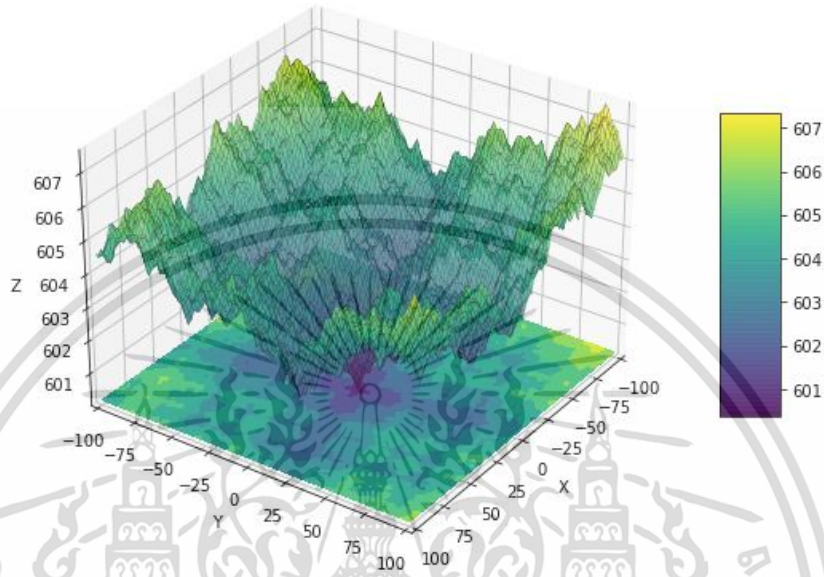


รูปที่ ก1.9 ฟังก์ชัน Shifted and Rotated Ackley ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Shifted and Rotated Weierstrass Function (F10) (CEC2014 no.6)

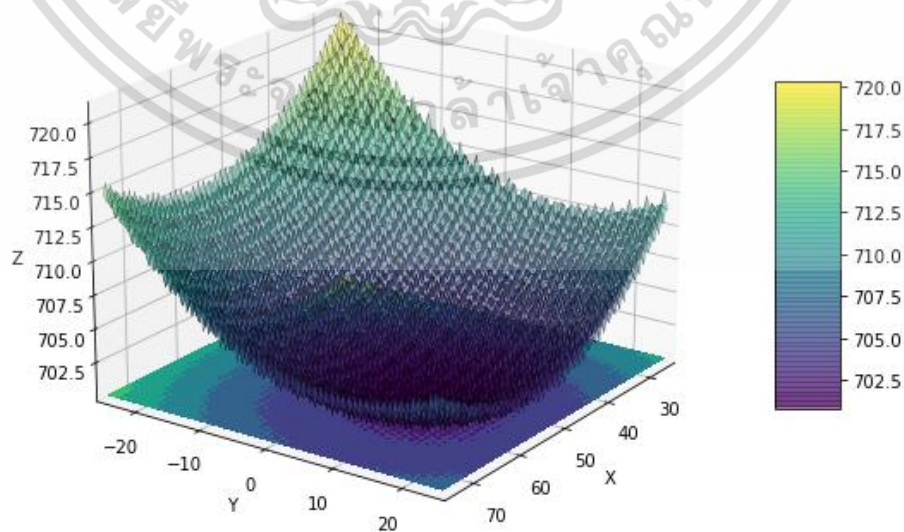
$$F_{10} = f_6 \left(M \left(\frac{0.5(x - o_6)}{100} \right) \right) + 600$$



รูปที่ ก1.10 ฟังก์ชัน Shifted and Rotated Weierstrass ในระนาบ 2 มิติ

11. Shifted and Rotated Griewank's Function (F11) (CEC2014 no.7)

$$F_{11} = f_7 \left(M \left(\frac{600(x - o_7)}{100} \right) \right) + 700$$

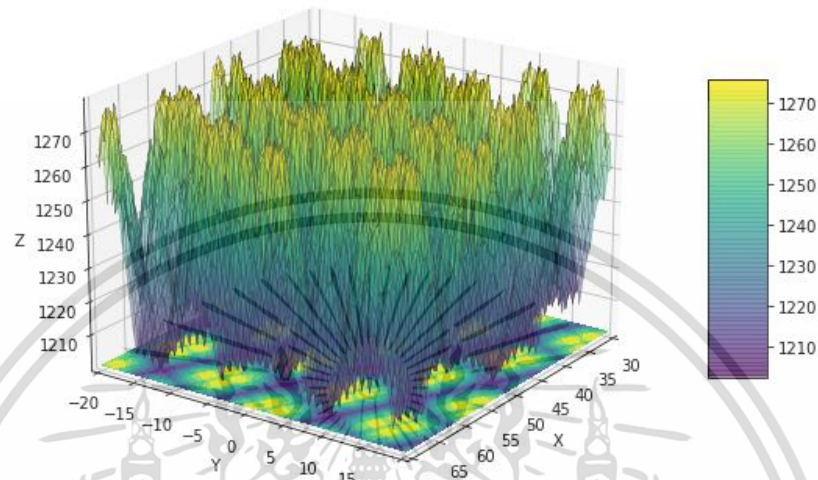


รูปที่ ก1.11 ฟังก์ชัน Shifted and Rotated Ackley ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. Shifted and Rotated Katsuura Function (F12) (CEC2014 no.12)

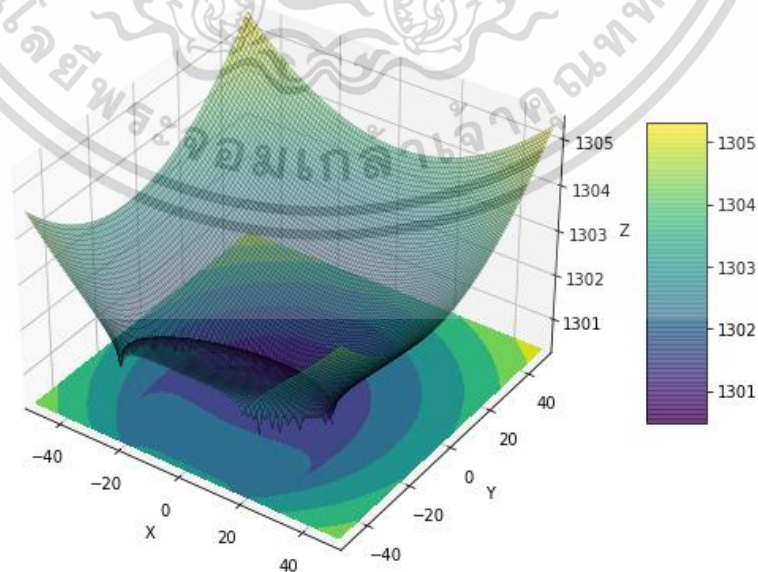
$$F_{12} = f_{10} \left(M \left(\frac{5(x - o_{12})}{100} \right) \right) + 1200$$



รูปที่ ก1.12 ฟังก์ชัน Shifted and Rotated Katsuura ในระนาบ 2 มิติ

13. Shifted and Rotated HappyCat Function (F13) (CEC2014 no.13)

$$F_{13} = f_{11} \left(M \left(\frac{5(x - o_{13})}{100} \right) \right) + 1300$$

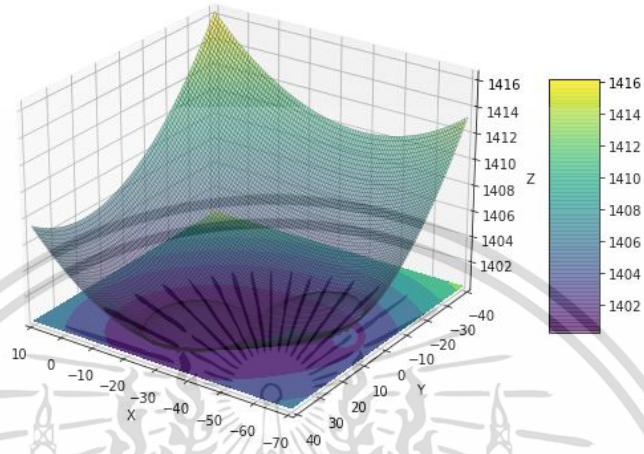


รูปที่ ก1.13 ฟังก์ชัน Shifted and Rotated Weierstrass ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. Shifted and Rotated HGBat Function (F14) (CEC2014 no.14)

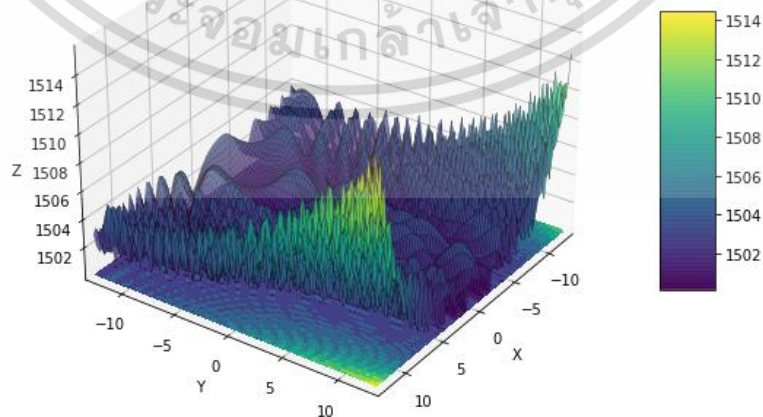
$$F_{14} = f_{12} \left(M \left(\frac{5(x - o_{14})}{100} \right) + 1 \right) + 1400$$



รูปที่ ก1.14 ฟังก์ชัน Shifted and Rotated HGBat ในระนาบ 2 มิติ

15. Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function (F15) (CEC2014 no.15)

$$F_{15} = f_{13} \left(M \left(\frac{5(x - o_{15})}{100} \right) + 1 \right) + 1500$$

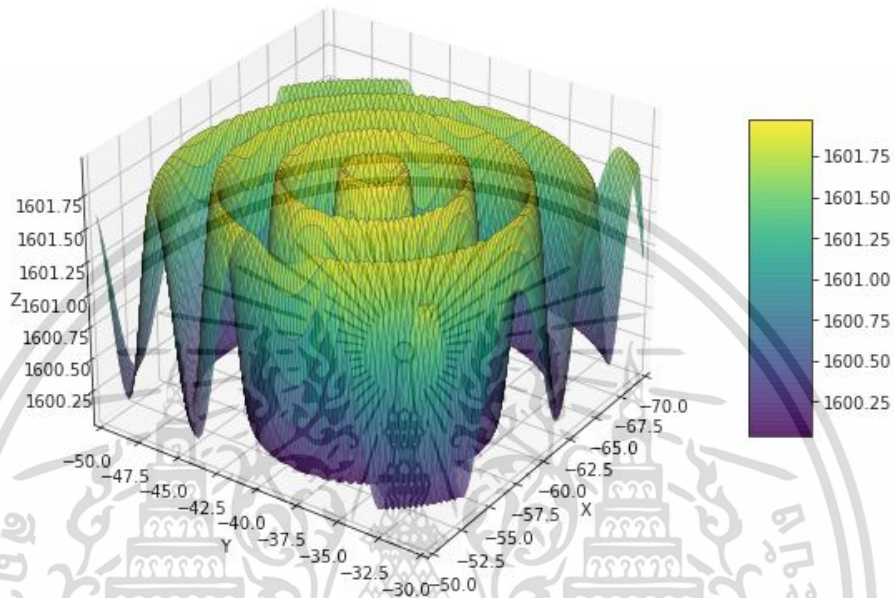


รูปที่ ก1.15 ฟังก์ชัน Shifted and Rotated Expanded Griewank plus Rosenbrock ในระนาบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16. Shifted and Rotated Expanded Scaffer's F6 Function (F16)
(CEC2014 no.16)

$$F_{16} = f_{14}(M(x - o_{16}) + 1) + 1600$$



รูปที่ ก1.16 ฟังก์ชัน Shifted and Rotated Expanded Scaffer ในระนาบ 2 มิติ

17. CEC2014 no.17 (F17)

$$F_{17} = \begin{pmatrix} 0.3f_9(M(x - o)) \\ +0.3f_8(M(x - o)) \\ +0.4f_1(M(x - o)) \end{pmatrix}$$

18. CEC2014 no.18 (F18)

$$F_{18} = \begin{pmatrix} 0.3f_2(M(x - o)) \\ +0.3f_{12}(M(x - o)) \\ +0.4f_8(M(x - o)) \end{pmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19.CEC2014 no.19 (F19)

$$F_{19} = \begin{pmatrix} 0.2f_7(M(x - o)) + 0.2f_6(M(x - o)) \\ 0.3f_4(M(x - o)) + 0.3f_{14}(M(x - o)) \end{pmatrix}$$

20.CEC2014 no.20 (F20)

$$F_{19} = \begin{pmatrix} 0.2f_{12}(M(x - o)) + 0.2f_3(M(x - o)) \\ 0.3f_{13}(M(x - o)) + 0.3f_8(M(x - o)) \end{pmatrix}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

โจทย์ปัญหาการเคลื่อนที่ในการทดลอง (Motion Problem Set)

ปัญหาทั้งจะเป็นปัญหาการเคลื่อนที่ใน 2 แนวแกน มีทั้งหมด 10 ปัญหาด้วยกันดังนี้

1. Point [4] = {{137,134} , {178,205} , {276,184} , {296,119}}
2. Point [5] = {{273,150} , {186,148} , {269,230} , {368,153} , {270,71}}
3. Point [7] = {{197,63} , {237,181} , {270,63} , {285,180} , {326,63} , {157,123} , {342,149}}
4. Point [8] = {{217,152} , {267,96} , {333,92} , {371,138} , {335,233} , {265,237} , {218,194} , {290,203}}
5. Point [8] = {{273,74} , {416,216} , {230,149} , {439,102} , {287,224} , {349,46} , {362,220} , {306,56}}

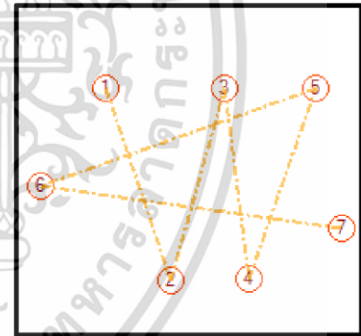
Problem No.1



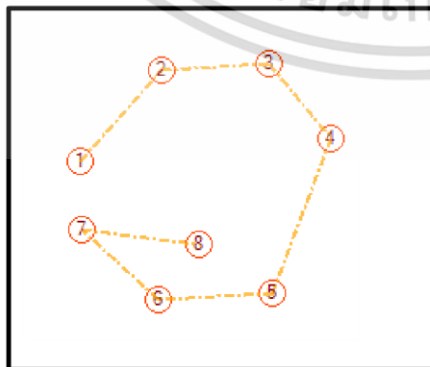
Problem No.2



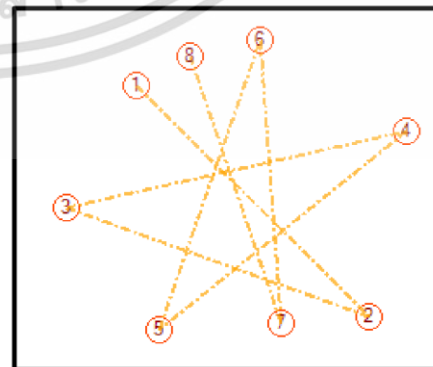
Problem No.3



Problem No.4



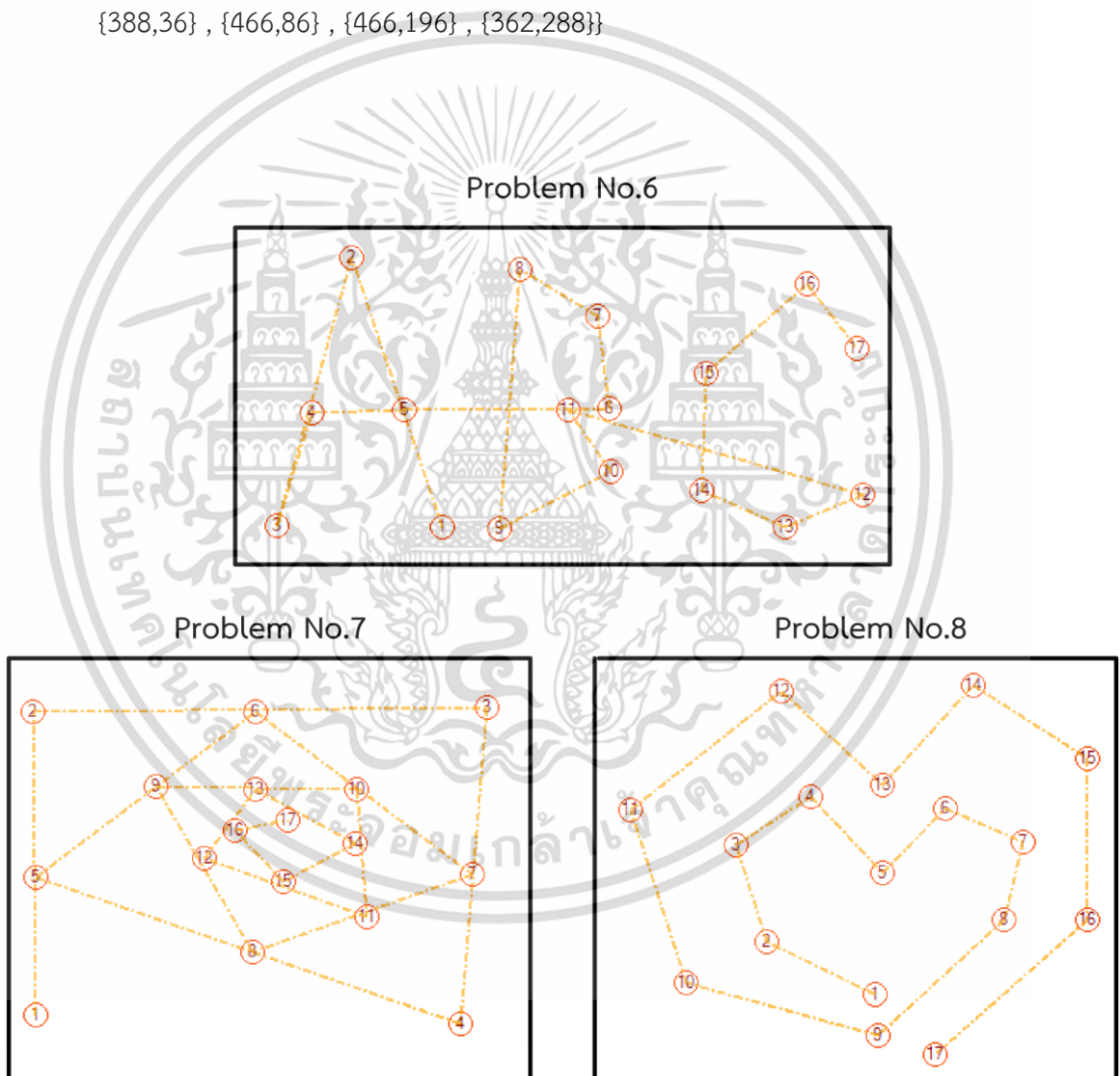
Problem No.5



รูปที่ ข1.1 ปัญหาการเคลื่อนที่ลำดับที่ 1 ถึง 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Point [17] = {{239,244} , {177,60} , {126,243} , {150,166} , {213,164} , {353,163} , {345,100} , {292,68} , {278,245} , {354,206} , {325,164} , {526,222} , {473,244} , {416,219} , {419,139} , {488,78} , {522,122}}
7. Point [17] = {{174,246} , {172,39} , {482,36} , {464,252} , {174,152} , {324,39} , {472,150} , {322,203} , {256,90} , {393,92} , {400,179} , {289,139} , {324,92} , {392,129} , {343,155} , {310,120} , {346,113}}
8. Point [17] = {{321,247} , {247,211} , {226,145} , {277,112} , {326,164} , {369,120} , {422,143} , {409,196} , {323,275} , {192,239} , {154,121} , {257,40} , {326,104} , {388,36} , {466,86} , {466,196} , {362,288}}

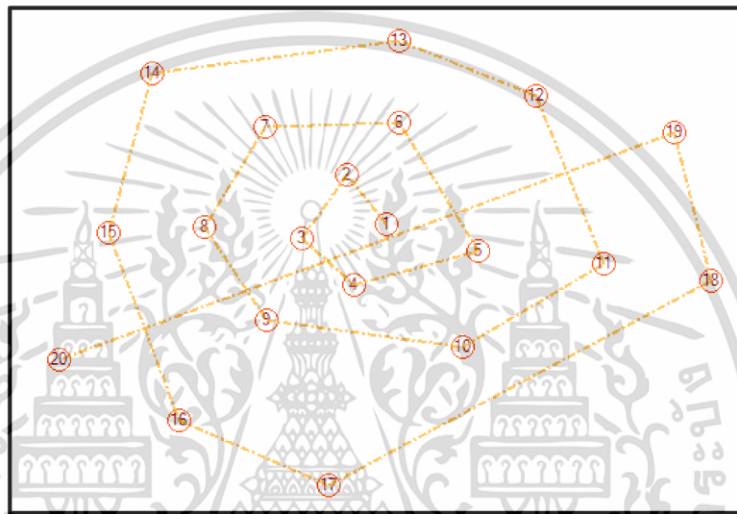


รูปที่ ข1.2 ปัญหาการเคลื่อนที่ลำดับที่ 6 ถึง 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Point [20] = {{409,162} , {381,127} , {349,172} , {386,205} , {474,181} , {418,90} ,
 {323,93} , {280,164} , {324,230} , {463,249} , {563,190} , {515,71} , {418,32} ,
 {243,55} , {212,168} , {262,301} , {368,347} , {639,202} , {613,97} , {177,258}}
10. Point [20] = {{131,50} , {125,225} , {194,61} , {147,128} , {204,229} , {280,70} ,
 {296,208} , {340,69} , {382,217} , {501,226} , {443,198} , {436,92} , {386,65} ,
 {486,64} , {672,78} , {574,97} , {588,221} , {740,79} , {765,222} , {888,224}}

Problem No.9



Problem No.10



รูปที่ ข1.3 ปัญหาการเคลื่อนที่ลำดับที่ 9 และ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.
ผลงานที่ได้รับการตีพิมพ์

1. S. Thongkrait and V. Chutchavong, "A Time Improvement PSO Base Algorithm Using Multithread Programming," 2019 4th International Conference on Communication and Information Systems (ICCIS), Wuhan, China, 2019, pp. 212-216, doi: 10.1109/ICCIS49662.2019.00044.
2. S. Thongkrait and V. Chutchavong, "Hard Deterministic Particle Swarm Optimisation for Certain Result Solution" International Conference on Electronics, Information, and Communication 2021. IEEE, 2021, pp. 629-632
3. Somsin Thongkrait, Vanvisa Chutchavong. "A Novel Evolution Optimization Algorithm Using a Multidimensional Geometric Method: Pivot Optimiser" International Journal of Simulation- Systems, Science and Technology N4 (2020), doi: 10.5013/IJSSST.a.21.04.13

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายสมสิน ทองไกรรัตน์
วัน เดือน ปีเกิด	28 กรกฎาคม 2535 ที่ขอนแก่น
ที่อยู่	804 หมู่ 1 ต.ปะหลาน อ.พยัคฆภูมิพิสัย จ.มหาสารคาม 44110 โทร 0897143534
ประวัติการศึกษา	2557 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2559 วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) โครงสร้างข้อมูลและอัลกอริทึม 2.) ระบบสมองกลแบบฝังตัว 3.) การหาค่าที่เหมาะสมที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้