

วิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดในโลคอลออปติ멈
ของพีเอสโอ

MERGE AND SPLIT PARTICLE SWARM IN REDUCING TRAPPING IN
LOCAL OPTIMA OF PSO



ลัคนา เพิ่มพูล
LUKKANA POEMPOOL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2561

KMITL-2018-EN-M-070-159

วิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดในโลคอลออปติ่มม
ของพีเอสโอ

MERGE AND SPLIT PARTICLE SWARM IN REDUCING TRAPPING IN
LOCAL OPTIMA OF PSO



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติ ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และข้อมูลอื่นใดถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พ.ศ.2561

KMITL-2018-EN-M-070-159

MERGE AND SPLIT PARTICLE SWARM IN REDUCING TRAPPING IN
LOCAL OPTIMA OF PSO



LUKKANA POEMPOOL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MAETER OF ENGINEERING IN COMPUTER ENGINEERING
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้ในเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้งานด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2018
KMUTL-2018-EN-M-070-159



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
FACULTY OF ENGINEERING
ไม่ว่ากรณีใด ๆ ทั้งสิ้น ยกเว้นที่มีเหตุผลแบบสงวนเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	วิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดในโลคอล ออปติ멈ของพีเอสโอ
นักศึกษา	นางสาวลัคนา เพิ่มพูล
รหัสนักศึกษา	58601337
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2561
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.บุญธีร์ เครือตราชู

บทคัดย่อ

จากปัญหาการติดใน Local Optima ของวิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO) ทำให้อนุภาคไม่สามารถทำการค้นหาค่าค่าตอบที่ดีที่สุดได้ วิทยานิพนธ์ฉบับนี้จึงได้ทำการพัฒนาวิธีการเพื่อแก้ปัญหาดังกล่าว ด้วยการทำการกลายพันธุ์ตำแหน่งของอนุภาค (particles) ตำแหน่งของอนุภาคที่ดีที่สุด (GBEST) และการใช้ตำแหน่งของอนุภาคที่ดีที่สุดจากกลุ่มอื่น มาใช้ร่วมกับการรวมและกระจายกลุ่มของอนุภาค ซึ่งเป็นการใช้ความสามารถในการค้นหาแบบเฉพาะที่ (Local Search) และแบบกว้าง (Global Search) ได้เหมาะสม โดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพในการค้นหาค่าตอบของ PSO วิทยานิพนธ์นี้จึงได้นำเสนอวิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดใน Local Optima ของ PSO โดยทำการทดสอบกับฟังก์ชันมาตรฐาน 27 ฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	MERGE AND SPLIT PARTICLE SWAM IN REDUCTING TRAPPING IN LOCAL OPTIMA OF PSO
Student	Ms. Lukkana Poempool
Student ID.	58601337
Degree	Master of Engineering
Program	Computer Engineering
Year	2018
Thesis Advisor	Assoc. Prof. Dr. Boontee Kruatrachue

ABSTRACT

The issue of Particle Swarm Optimization (PSO) is particles move into the local optima, which is why particles cannot find the best solution. This thesis is developed by mutate positions of particle and GBEST or use GBEST from other groups, in addition, its merge and split particle swarm to balance ability of local search and global search. The proposed algorithm is Merge and Split particle swarm in reducing trapping in local optima of PSO (MSPSO). This algorithm tested on 27 benchmark test functions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษา รศ.ดร.บุญธีร์ เครือตราชู ที่ให้ความช่วยเหลือ ให้คำชี้แนะแนวทางในการเรียนรู้ ช่วยแก้ปัญหา ตลอดจนให้ข้อคิดและประสบการณ์ที่ดี จนกระทั่งแล้วเสร็จ

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะเพื่อให้เนื้อหาของงานวิจัยครอบคลุมและครบถ้วนจนทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์และสำเร็จลงได้

ขอขอบคุณเจ้าหน้าที่และอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ ที่คอยให้ความช่วยเหลือต่างๆ ตลอดจนนักวิจัยทุกท่านที่เอื้อเฟื่องานวิจัยจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

ขอขอบคุณพี่ น้อง และเพื่อน ระดับปริญญาโท รวมถึงเพื่อนร่วมห้องวิจัยที่คอยให้ความช่วยเหลือในเรื่องต่างๆจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง ตลอดจนสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่อำนวยความสะดวกในการทำงานวิจัย

ขอขอบคุณเพื่อนๆของข้าพเจ้า ที่เป็นที่ปรึกษา รับฟัง และเป็นกำลังใจให้ข้าพเจ้าตลอดมา สุดท้ายต้องขอขอบคุณ คุณชรินทร์ เพิ่มพูล และคุณดลยา เพิ่มพูล บิดามารดาของข้าพเจ้าที่ให้โอกาสและสนับสนุนข้าพเจ้าตลอดมา ขอขอบคุณ คุณรติ เพิ่มพูล พี่ชายของข้าพเจ้าที่เป็นที่ปรึกษาที่ดี ขอขอบคุณครอบครัวของข้าพเจ้าที่เป็นกำลังใจที่สำคัญยิ่งตลอดการเรียนและการทำงานวิจัยเล่มนี้

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้บิดามารดาซึ่งเป็นที่ยรักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า หวังว่าผู้อ่านคงจะได้ความรู้จากวิทยานิพนธ์เล่มนี้ไม่มากนักน้อย หากมีข้อผิดพลาดประการใดข้าพเจ้าขอน้อมรับไว้แต่เพียงผู้เดียว

ลัดดา เพิ่มพูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VIII

บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 จุดมุ่งหมายและวัตถุประสงค์.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ขอบเขตการวิจัย.....	3
1.5 ขั้นตอนการศึกษา.....	3
1.6 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย.....	3
1.7 โครงสร้างของวิทยานิพนธ์.....	3

บทที่ 2 ทฤษฎีพื้นฐานที่เกี่ยวข้อง	5
2.1 ปัญหาการหาค่าที่เหมาะสมที่สุด	5
2.2 ปัญหาการติดในจุดที่ดีที่สุดเฉพาะที่.....	6
2.3 วิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาค	7
2.4 รูปแบบในการค้นหาคำตอบ	9

บทที่ 3 งานวิจัยที่เกี่ยวข้อง	11
3.1 วิธีการค้นหาค่าความเหมาะสมแบบหลายกลุ่มอนุภาค	11
3.2 วิธีการค้นหาอย่างรวดเร็วแบบหลายกลุ่มอนุภาคโดยการกลายพันธุ์และการผสมพันธุ์. 11	
3.3 วิธีการค้นหาขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบ หลากหลายกลุ่มโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ร่วมกับการ เปลี่ยนตำแหน่งของอนุภาคที่ดีที่สุดของกลุ่ม.....	15

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรืออ้างถึงในกรณี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่สิ่งนี้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มาใช้

สารบัญ (ต่อ)

	หน้า
3.4 ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุ่มสัมพัทธ์	18
3.5 วิธีการเริ่มต้นการทำงานใหม่ของการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาคด้วยการกลายพันธุ์ของตำแหน่งที่ดีที่สุดของอนุภาค.....	19
บทที่ 4 การปรับปรุงการกลายพันธุ์และรีโพอิชัน.....	20
บทที่ 5 ผลการทดลอง.....	31
5.1 ค่าพารามิเตอร์	32
5.2 การทดลอง.....	32
5.3 ฟังก์ชันที่ใช้ในการทดลอง.....	33
5.4 ผลการทดลอง	34
บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ	44
6.1 สรุปผลการทดลอง	44
6.2 ข้อเสนอแนะ	45
เอกสารอ้างอิง	46
ภาคผนวก.....	48
ภาคผนวก ก. รายละเอียดของฟังก์ชันมาตรฐาน	49
ภาคผนวก ข. งานวิจัยที่ได้รับการตีพิมพ์	69
ภาคผนวก ค. การปรับเปลี่ยนค่าพารามิเตอร์ที่เหมาะสมของวิธี MSPSO.....	75
ภาคผนวก ง. การเลื่อนตำแหน่งโดเมนของฟังก์ชัน.....	86

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรเอาไปใช้ประโยชน์ด้านอื่น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
5.1 ค่าพารามิเตอร์พื้นฐานที่ใช้ในการทดลองกับฟังก์ชันมาตรฐาน.....	32
5.2 ลักษณะและจำนวนมิติของฟังก์ชัน.....	33
5.3 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานแบบ 1 กลุ่ม อนุภาค PSO, RPSO, GBPSO และที่นำเสนอที่เป็นแบบหลายกลุ่มอนุภาค CMPSO กับMSPSO ...	34
5.4 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานแบบ 1 กลุ่ม อนุภาค PSO, RPSO, GBPSO และที่นำเสนอที่เป็นแบบหลายกลุ่มอนุภาค CMPSO กับMSPSO (ต่อ)	35
5.5 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานหลายกลุ่มอนุภาค FMP SO, MPSOSM, และ CMPSO กับอัลกอริทึมที่นำเสนอ (MSPSO)	38
5.6 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานหลายกลุ่มอนุภาค FMP SO, MPSOSM, และ CMPSO กับอัลกอริทึมที่นำเสนอ (MSPSO) (ต่อ)	39
ก.1 สรุปรายละเอียดพื้นฐานของฟังก์ชันที่ใช้ในการทดลอง.....	68
ค.1 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของอนุภาค.....	76
ค.2 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของอนุภาค(ต่อ).....	77
ค.3 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของGBEST	78
ค.4 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของGBEST(ต่อ).....	79
ค.5 ตารางแสดงผลการทดสอบจำนวนมิติที่ใช้ในการทดลอง.....	80
ค.6 ตารางแสดงผลการทดสอบค่า Local Optima Threshold.....	81
ค.7 ตารางแสดงผลการทดสอบจำนวนกลุ่มที่ใช้ในการแบ่งกลุ่มอนุภาค	83
ค.8 ตารางแสดงผลการทดสอบการปรับเปลี่ยนจำนวนครั้งในการกระจายอนุภาคแบบหลายกลุ่ม อนุภาค และแบบ 1 กลุ่มอนุภาค.....	84
ง.1 ตารางแสดงผลการค้นหาค่าคำตอบของวิธี MSPSO โดยทำการเปลี่ยนตำแหน่งของค่าคำตอบที่ ดีที่สุดไปยังตำแหน่งอื่นๆ.....	86
ง.2 ตารางแสดงผลการค้นหาค่าคำตอบของวิธี MSPSO โดยทำการเปลี่ยนตำแหน่งของค่าคำตอบที่ ดีที่สุดไปยังตำแหน่งอื่นๆ(ต่อ).....	87
ง.3 ตารางแสดงผลการเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่ง 25%ของขอบเขตบน ของแต่ละฟังก์ชันเพื่อเปรียบเทียบประสิทธิภาพระหว่าง GBPSO 30 อนุภาค และ 90 อนุภาค กับ การค้า ไม่ว่าการค้าใน MSPSO.....	89

ง.4 ตารางแสดงผลการเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่งอื่นๆแบบสุ่มโดยที่คำตอบในแต่ละมิติมีค่าที่ไม่เท่ากัน(Random Shift) เพื่อเปรียบเทียบประสิทธิภาพระหว่าง GBPSO 30 อนุภาค และ 90 อนุภาค กับ MSPSO..... 90



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ฟังก์ชัน ACKLEY	5
2.2 ตัวอย่างของปัญหา Unimodal และ Multi-modal.....	6
2.3 แผนผังการทำงานของ PSO.....	7
2.4 รูปแบบในการค้นหาคำตอบ.....	10
3.1 แผนผังการทำงานของ FPSO	13
3.2 แผนผังการทำงานของ FMPSO	15
3.3 แผนผังการทำงานของ MPSOSM.....	17
3.4 แผนผังการทำงานของ RPSO	18
3.5 แผนผังการทำงานของ GBPSO.....	19
4.1 กราฟ PSO แสดงพฤติกรรมการค้นหาคำตอบของฟังก์ชัน ROSENBROCK.....	20
4.2 กราฟเปรียบเทียบการติด Local Optima ระหว่าง PSO กับ MPSO ในการค้นหาคำตอบของฟังก์ชัน ROSENBROCK.....	21
4.3 แผนผังการทำงานของ CMPSO.....	22
4.4 กราฟแสดงการทดสอบการแบ่งจำนวนกลุ่มอนุภาคเพื่อทดสอบ CMPSO.....	23
4.5 รหัสจำลองแสดงการทำงานของ MSPSO	27
4.6 รหัสจำลองแสดงการทำงานของ MSPSO(ต่อ).....	28
4.7 แผนผังการทำงานของ MSPSO	30
5.1 กราฟแสดงการค้นหาคำตอบบนฟังก์ชัน ROSENBROCK ของอัลกอริทึม PSO, RPSO, GBPSO, FMPSO, MPSOSM, CMPSO และ MSPSO	40
5.2 กราฟแสดงการค้นหาคำตอบบนฟังก์ชัน SCHWEFEL ของอัลกอริทึม PSO, RPSO, GBPSO, FMPSO, MPSOSM, CMPSO และ MSPSO.....	42
ก.1 ฟังก์ชัน ACKLEY	50
ก.2 ฟังก์ชัน GRIEWANK.....	50
ก.3 ฟังก์ชัน RASTRIGIN	51
ก.4 ฟังก์ชัน ROSENBROCK	52
ก.5 ฟังก์ชัน SCHWEFEL.....	52
ก.6 ฟังก์ชัน COSINE MIXTURE.....	53
ก.7 ฟังก์ชัน EXPONENTIAL.....	54
ก.8 ฟังก์ชัน LEVY	54

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.9 ฟังก์ชัน MICHALEWICZ	55
ก.10 ฟังก์ชัน DIXON-PRICE.....	56
ก.11 ฟังก์ชัน STEP	56
ก.12 ฟังก์ชัน SCHAFFER	57
ก.13 ฟังก์ชัน HOLDER	58
ก.14 ฟังก์ชัน BEALE.....	58
ก.15 ฟังก์ชัน SHUBERT	59
ก.16 ฟังก์ชัน GOLDSTEIN-PRICE	60
ก.17 ฟังก์ชัน SIX HUMP CAMEL	60
ก.18 ฟังก์ชัน SPHERE	61
ก.19 ฟังก์ชัน PARALLEL ELLIPSOID	62
ก.20 ฟังก์ชัน ROTATED ELLIPSOID	62
ก.21 ฟังก์ชัน CIGAR	63
ก.22 ฟังก์ชัน BROWN	64
ก.23 ฟังก์ชัน MULTIMOD.....	64
ก.24 ฟังก์ชัน ZAKHAROV.....	65
ก.25 ฟังก์ชัน TRID.....	66
ก.26 ฟังก์ชัน EASOM.....	66
ก.27 ฟังก์ชัน EEGHOLDER	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การค้นหาค่าที่เหมาะสมที่สุด (optimization) เป็นกระบวนการสำคัญที่นำไปประยุกต์ใช้ในการออกแบบงานทางด้านวิศวกรรม วิทยาศาสตร์ อุตสาหกรรม เศรษฐศาสตร์ ธุรกิจ และการขนส่ง เป็นต้น เพื่อให้ได้ผลลัพธ์ตามที่ต้องการและมีประสิทธิภาพสูงสุด อัลกอริทึมที่นิยมนำมาใช้ในการแก้ปัญหาคือ Meta-heuristic algorithm เนื่องจากเป็นอัลกอริทึมที่ใช้หลักการในการสุ่มเดาคำตอบเพื่อทำการค้นหาค่าที่เหมาะสมที่สุดซึ่งเหมาะกับปัญหาที่มีความยุ่งยากซับซ้อน และหนึ่งในวิธีที่เป็นที่นิยมนำมาใช้ในการหาค่าความเหมาะสมก็คือ วิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization, PSO) [1,2]

วิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาค หรือ PSO เป็นอัลกอริทึมที่ Kennedy และ Eberhart คิดค้นขึ้นในปี 1995 ทั้งคู่ได้เริ่มศึกษาและพัฒนาด้วยการพิจารณาจากพฤติกรรมทางสังคมของสัตว์ในการเคลื่อนที่เพื่อหาอาหารเป็นฝูง เช่น ฝูงนก หรือฝูงปลา เป็นต้น เนื่องจากสัตว์เหล่านี้มีพฤติกรรมในการออกหาอาหารพร้อมกันเป็นฝูง และการสื่อสารแลกเปลี่ยนข้อมูลระหว่างกัน ทั้งนี้ทุกตัวจะแทนด้วยอนุภาคและเคลื่อนที่อยู่บนขอบเขตของคำตอบ (search space) แต่ละอนุภาคจะทำการคำนวณค่าความเหมาะสม (Fitness value) ของแต่ละตำแหน่งเพื่อทำการเคลื่อนที่ โดยตามพฤติกรรมแล้วอนุภาคจะทำการเคลื่อนที่เข้าหาตำแหน่งที่ดีที่สุดหรือ GBEST โดยไม่ทำการกระจายตัวออกไปค้นหาที่บริเวณอื่นอีก แต่หากระหว่างทางในการเคลื่อนที่ไปหา GBEST อนุภาคตัวนั้นเจอค่าตำแหน่งที่ดีกว่าอนุภาคตัวนั้นก็จะเป็น GBEST แทน และอนุภาคตัวอื่นจะทำการเปลี่ยนทิศทางการเคลื่อนที่ไปหา GBEST ตัวใหม่ ฉะนั้นหาก GBEST เคลื่อนที่ไปติดในจุดที่ดีที่สุดเฉพาะที่ (Local Optima) ทุกอนุภาคจะทำการเคลื่อนที่มารวมตัวกันที่ตำแหน่งนั้นจนมีผลให้ค่าคำตอบที่ได้จะไม่ดีขึ้นไม่ว่าจะทำการค้นหาไปอีกกี่รอบ

ด้วยปัญหาที่กล่าวมาข้างต้นจึงได้มีนักวิจัยหลายท่านได้ทำการปรับปรุง พัฒนาและแก้ปัญหา

PSO โดยนำแนวคิดในการเพิ่มความสามารถในการค้นหาในแนวกว้าง (Global Search) ด้วยการแบ่งกลุ่มอนุภาคออกเป็นหลายกลุ่มย่อยแต่ละกลุ่มย่อยมีค่า GBEST เป็นของตัวเองหรือเรียกว่าวิธีการหาค่าความเหมาะสมแบบหลายกลุ่มอนุภาค (Multi - Particle Swarm Optimization,

MPSO) แต่เนื่องจากค่าคำตอบที่ได้จาก MPSO นี้ไม่ดีกว่าค่าคำตอบที่ได้จาก PSO ปกติ นักวิจัยจึงได้นำเทคนิคต่างๆมาปรับปรุงร่วมกับ MPSO เพื่อเพิ่มประสิทธิภาพในการค้นหา ดังเช่นวิธีการค้นหาอย่างรวดเร็วแบบหลากหลายกลุ่มโดยการกระจายพันธุ์และการผสมพันธุ์ (Fast Multi-swarm Particle Swarm Optimization : FMPSO) และขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบหลากหลายกลุ่มโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ร่วมกับการเปลี่ยนตำแหน่งของอนุภาคที่ดีที่สุดของกลุ่ม (Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm : MPSOSM) และอื่นๆ งานวิจัยเหล่านี้สามารถหาค่าคำตอบที่ดีขึ้นเมื่อเทียบกับ PSO ต้นแบบ แต่ก็ยังคงใช้เวลาในการค้นหาค่าคำตอบนาน

งานวิจัยนี้จึงได้นำเสนอวิธีการหาค่าที่เหมาะสมที่สุดแบบหลายกลุ่มอนุภาคร่วมกับการกระจายพันธุ์ตำแหน่งของ GBEST และอนุภาค หรือเลือกใช้ค่า GBEST จากกลุ่มอื่น เพื่อแก้ปัญหาการติดใน Local Optimum ของ PSO พร้อมทั้งทำการเพิ่มและลดจำนวนกลุ่มอนุภาคในช่วงเวลาที่เหมาะสมเพื่อเพิ่มประสิทธิภาพในการค้นหาค่าคำตอบที่ดีที่สุด

1.2 จุดมุ่งหมายและวัตถุประสงค์

วิทยานิพนธ์ฉบับนี้มีจุดมุ่งหมายและวัตถุประสงค์เพื่อศึกษาและนำเสนออัลกอริทึมที่สามารถแก้ปัญหาการหาค่าที่เหมาะสมที่สุด โดยให้ความสำคัญกับการแก้ปัญหาการติดใน Local Optima ของอนุภาคด้วยการทำการค้นหาแบบ PSO ควบคู่ไปกับการเพิ่มและลดจำนวนกลุ่มอนุภาคในช่วงเวลาที่เหมาะสมทั้งนี้ได้นำการรีโพลิชตำแหน่งของอนุภาคและ GBEST รวมถึงการอ้างอิงค่า GBEST จากกลุ่มอื่นมาประยุกต์ใช้ด้วย

1.3 สมมติฐานของการศึกษา

การทำการค้นหาค่าคำตอบที่เหมาะสมที่สุดแบบกลุ่มอนุภาคเดียวสลับกับการค้นหาแบบหลายกลุ่มอนุภาคร่วมกับทำการกระจายพันธุ์ตำแหน่งของอนุภาคและ GBEST หรือเลือกใช้ค่า GBEST จากกลุ่มอื่นเพื่อหลีกเลี่ยงการติดใน Local Optima และใช้ความสามารถในการค้นหาแบบ

เอกสารนี้เป็น Local Search ควบคู่ไปกับการค้นหาแบบ Global Search นั้นมีผลให้สามารถค้นหาค่าคำตอบที่ดีขึ้นและใช้เวลาในการค้นหาน้อยลง ไม่ว่ากรณีใดก็ตาม และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์ฉบับนี้ทำการกำหนดขอบเขตของงานวิจัยดังต่อไปนี้

1.5.1 เพื่อแก้ปัญหาการติดใน Local optima ของ PSO

1.5.2 เพื่อวัดประสิทธิภาพของวิธีการที่นำเสนอเกี่ยวกับอัลกอริทึม PSO ในรูปแบบต่าง ๆ โดยทำการทดลองกับฟังก์ชันมาตรฐาน

1.5 ขั้นตอนการศึกษา

ขั้นตอนการศึกษาของงานวิจัยนี้สามารถอธิบายได้ดังต่อไปนี้

1.5.1 ศึกษาทฤษฎีและความรู้พื้นฐานของวิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาค

1.5.2 ศึกษางานวิจัยที่เกี่ยวข้องกับ PSO ที่มีผู้พัฒนาและนำเสนอมาแล้ว

1.5.3 ศึกษาปัญหาฟังก์ชันมาตรฐานที่นำมาใช้ในการทดลอง

1.5.4 พัฒนาและออกแบบอัลกอริทึมเพื่อแก้ปัญหา

1.5.5 ทดสอบอัลกอริทึม กับอัลกอริทึม PSO ในรูปแบบต่าง ๆ

1.5.6 วิเคราะห์ผลลัพธ์ และแก้ไขปัญหาที่เกิดขึ้น

1.5.7 ทำการพัฒนาอัลกอริทึม และทำการทดสอบอีกครั้ง

1.5.8 สรุปผลการทดลองพร้อมวิเคราะห์ผลการทดลองที่ได้

1.5.9 จัดทำวิทยานิพนธ์

1.6 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

1.6.1 เครื่องคอมพิวเตอร์โน้ตบุ๊ก หน่วยประมวลผลกลาง Intel Core i5 หน่วยความจำหลัก (RAM) 8 GB จำนวน 1 เครื่อง

1.6.2 ระบบปฏิบัติการ Windows 8.1

1.6.3 Microsoft Visual C++ 2017

1.7 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท ภาคผนวก 4 บท แต่ละบทจะมีรายละเอียด
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ใดๆ
 ไม่ว่ากรณีใดๆทั้งสิ้น หากมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 ดังนี้

บทที่ 1 อธิบายถึงความเป็นมาของงานวิจัย จุดมุ่งหมายและวัตถุประสงค์ สมมติฐานของการศึกษา ขอบเขตของการวิจัย ขั้นตอนของการศึกษา เครื่องมือและอุปกรณ์ที่ใช้ในการวิจัย และโครงสร้างวิทยานิพนธ์

บทที่ 2 อธิบายถึงทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้อง ได้แก่ ปัญหาในการหาค่าความเหมาะสม วิธีการหาค่าความเหมาะสมแบบกลุ่มอนุภาค ข้อดีและข้อเสียของ PSO

บทที่ 3 อธิบายถึงงานวิจัยที่เกี่ยวข้อง ได้แก่ วิธีการค้นหาค่าความเหมาะสมแบบหลายกลุ่มอนุภาค (Multi Particle Swarm Optimization : MPSO) วิธีการค้นหาอย่างรวดเร็วแบบหลากหลายกลุ่มโดยการกลายพันธุ์และการผสมพันธุ์ (Fast Multi-swarm Particle Swarm Optimization : FMPSO) ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบหลากหลายกลุ่มโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ร่วมกับการเปลี่ยนตำแหน่งของอนุภาคที่ดีที่สุดของกลุ่ม (Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm : MPSOSM) ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ (Reposition by Slightly Mutation Particle with Particle Swarm Optimization : RPSO) และวิธีการเริ่มต้นการทำงานใหม่ของการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาคด้วยการกลายพันธุ์ของตำแหน่งที่ดีที่สุดของอนุภาค (The use of Global Best position in rerun of particle swarm optimization : GBPSO)

บทที่ 4 อธิบายถึงแนวคิด วิธี ที่นำมาใช้ในการปรับปรุงและแก้ปัญหา รวมถึงรายละเอียดของอัลกอริทึมที่นำเสนอ

บทที่ 5 อธิบายการทดลองโดยวิเคราะห์ผ่านฟังก์ชันมาตรฐาน 27 ฟังก์ชัน และเปรียบเทียบผลลัพธ์กับ PSO วิธีการอื่น ๆ

บทที่ 6 สรุปผลการทดลอง ข้อเสนอแนะ และแนวทางในการปรับปรุง

ภาคผนวก ก กล่าวถึงรายละเอียดของฟังก์ชันมาตรฐานที่นำมาใช้ในการทดลองในบทที่ 5

ภาคผนวก ข งานวิจัยที่ได้รับการตีพิมพ์

ภาคผนวก ค การทดลองปรับเปลี่ยนค่าพารามิเตอร์ที่เหมาะสมของวิธี MSPSO

ภาคผนวก ง การทดลองเลื่อนตำแหน่งโดเมนของฟังก์ชันมาตรฐานเพื่อวัดประสิทธิภาพในการค้นหาค่าคำตอบของวิธีที่นำเสนอ และ GBPSO(ทดลองเพิ่มเติม)

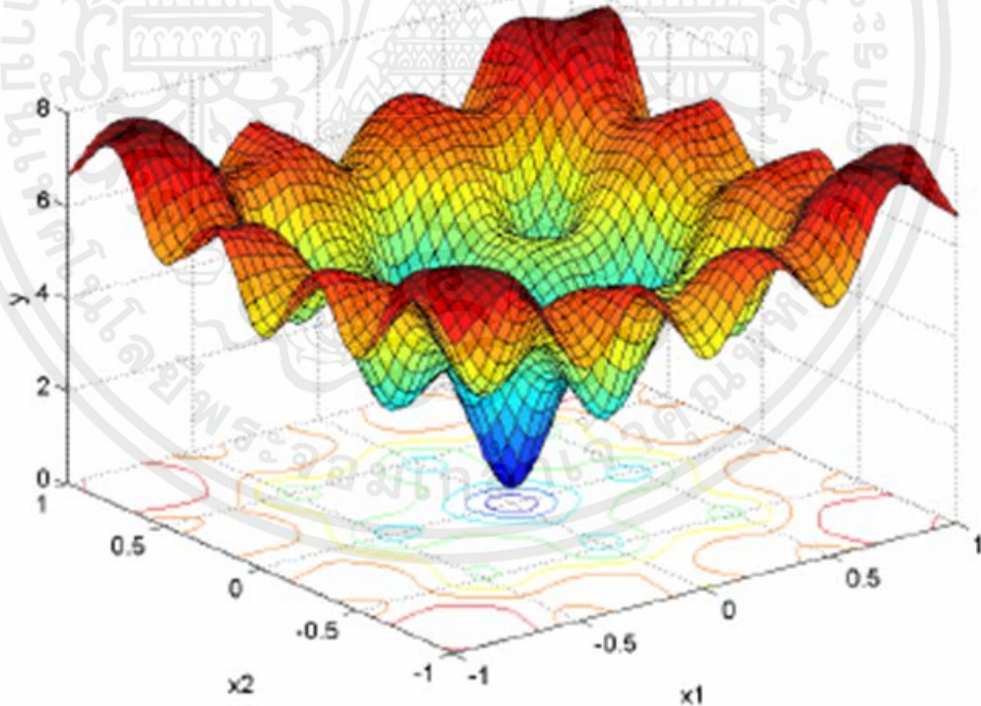
บทที่ 2

ทฤษฎีพื้นฐานที่เกี่ยวข้อง

2.1 ปัญหาการหาค่าที่เหมาะสมที่สุด (Optimization Problem)

ในทางคณิตศาสตร์และทางวิทยาศาสตร์คอมพิวเตอร์ได้ให้นิยามของปัญหาการหาค่าที่เหมาะสมที่สุดหรือปัญหาในการหาจุดที่ดีที่สุด [17] ไว้ว่า ปัญหาการหาค่าที่เหมาะสมที่สุด คือ ปัญหาของการหาค่าที่ดีที่สุดจากคำตอบที่เป็นไปได้ทั้งหมดหรือกล่าวอีกนัยหนึ่งก็คือ การหาค่าสูงสุด หรือ ค่าต่ำสุดของปัญหาหรือฟังก์ชัน โดยแสดงปัญหาอยู่ในรูปของสมการทางคณิตศาสตร์

ปัญหา หรือ ฟังก์ชัน ที่นำมาใช้ในการหาค่าสูงสุดหรือต่ำสุดนั้น เรียกว่า ฟังก์ชันวัตถุประสงค์ หรือ ฟังก์ชันเป้าหมาย (Objective function) ซึ่งในวิทยานิพนธ์ฉบับนี้ได้นำฟังก์ชันมาตรฐานมาใช้ในการวัดประสิทธิภาพ เช่น ฟังก์ชัน ACKLEY ดังรูป 2.1 สามารถแสดงปัญหาให้อยู่ในรูปของสมการทางคณิตศาสตร์ได้ดังสมการที่ 2.1



รูปที่ 2.1 ฟังก์ชัน ACKLEY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น คือทั้งห้ามมิให้คัดลอกและใช้อย่าง (1) จำนวนเอกสารทุกครั้งที่มีการนำไปใช้

$$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}\right) - \exp\left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i)\right) + 20 + e \quad (2.1)$$

โดยที่

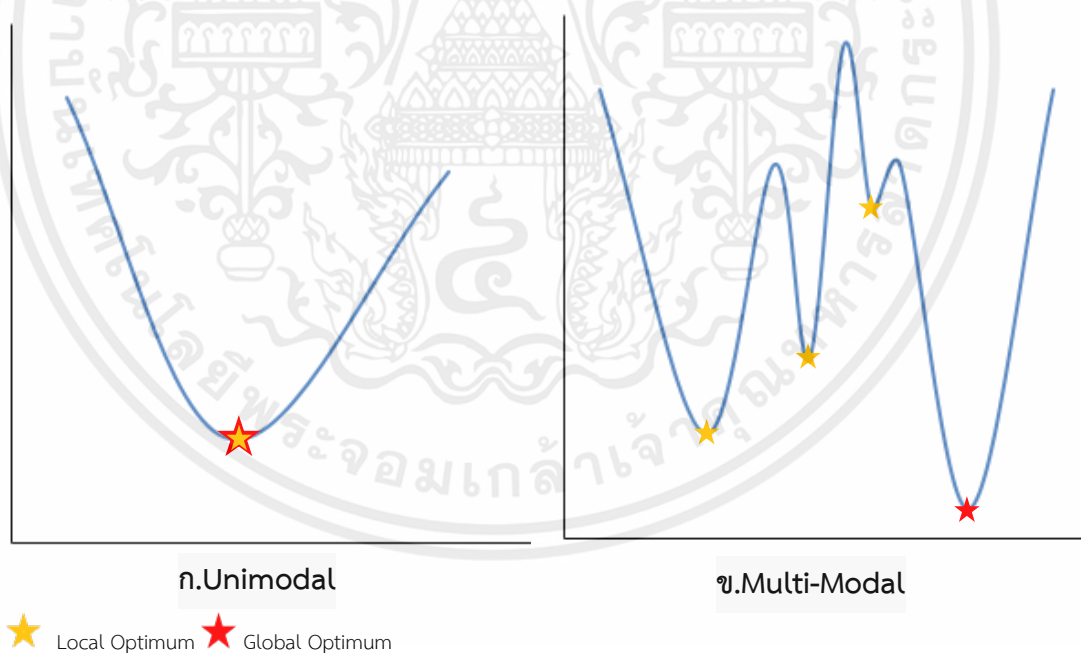
- $f(x)$: ฟังก์ชันเป้าหมาย
 x : ตัวแปรต้น (Control Variables) หรือ คำตอบของปัญหา
 d : จำนวนมิติของปัญหา
 i : มิติของปัญหา

จากรูปที่ 2.1 จะเห็นว่า ฟังก์ชัน ACKLEY เป็นฟังก์ชันเป้าหมายที่มีปัญหาในรูปแบบ Multi-modal เป้าหมายของปัญหาก็คือ การหาค่า x ที่ทำให้ $f(x)$ มีค่าน้อยที่สุด

ปัญหาการหาค่าที่เหมาะสมที่สุดแบ่งรูปแบบของปัญหาได้ 2 ลักษณะดังนี้

1. Unimodal problem คือ ปัญหาการหาค่าที่เหมาะสมที่สุด บนพื้นที่การค้นหาที่มีเพียงจุด Global optimum เพียงจุดเดียวเท่านั้น หรืออาจกล่าวได้ว่า จุด Global optimum และจุด Local optimum คือจุดเดียวกัน ดังรูปที่ 2.2 ก

2. Multi-modal problem คือ ปัญหาการหาค่าที่เหมาะสมที่สุด บนพื้นที่การค้นหาที่มีจุด Local optimum ตั้งแต่หนึ่งจุดขึ้นไป แต่มีจุด Global optimum เพียงจุดเดียวเท่านั้น ดังรูปที่ 2.2 ข



รูปที่ 2.2 ตัวอย่างของปัญหา Unimodal และ Multi-modal

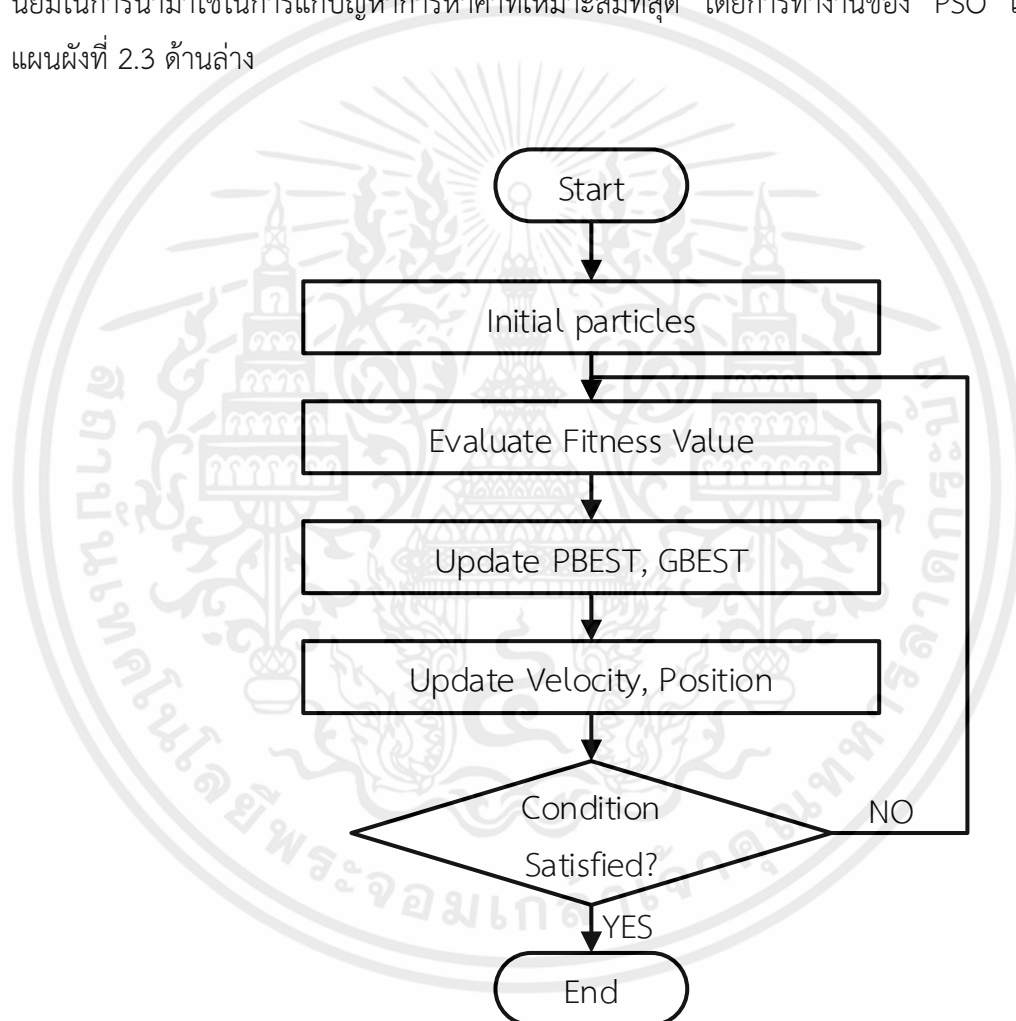
2.2 ปัญหาการติดในจุดที่ดีที่สุดเฉพาะที่ (Local Optimum Problem)

ปัญหาการติดในจุดที่ดีที่สุดเฉพาะที่ คือ กระบวนการค้นหาค่าที่เหมาะสมที่สุดนั้นไม่สามารถหาคำตอบที่ดีที่สุดของปัญหา หรือจุด Global optima (จุดที่เป็นค่าคำตอบที่ดีที่สุดของปัญหาจะมี

ได้เพียง 1 จุด ต่อหนึ่งปัญหาเท่านั้น) ได้ เนื่องจากอนุภาคทุกตัวติดอยู่ในค่าคำตอบเฉพาะที่ หรือจุด Local optima (จุดที่เป็นคำตอบของปัญหา อาจมีได้หลายจุดต่อหนึ่งปัญหา) ไม่สามารถทำการเคลื่อนที่ออกไปค้นหาบริเวณอื่นได้อีก

2.3 วิธีการหาค่าที่เหมาะสมที่สุดแบบอนุภาคกลุ่ม (Particle Swarm Optimization, PSO)

วิธีการหาค่าที่เหมาะสมที่สุดแบบอนุภาคกลุ่ม [1,2] หรือ PSO เป็นหนึ่งในอัลกอริทึมที่เป็นที่นิยมในการนำมาใช้ในการแก้ปัญหาการหาค่าที่เหมาะสมที่สุด โดยการทำงานของ PSO แสดงตามแผนผังที่ 2.3 ด้านล่าง



รูปที่ 2.3 แผนผังการทำงานของ PSO

เอกสารนี้เป็นเอกสารจากแผนผังการทำงานของ PSO ในรูปที่ 2.3 สามารถอธิบายถึงขั้นตอนการทำงานของ PSO คำ
ไม่ว่ากรณีใดได้ดังนี้ อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1

กำหนดค่าเริ่มต้นให้อนุภาค (Initial Particles) ด้วยการสุ่มตำแหน่งและความเร็วเริ่มต้นของอนุภาคทุกตัวลงใน search space ในขั้นตอนนี้จะทำเพียงครั้งเดียวเท่านั้น

ขั้นตอนที่ 2

การประเมินค่าคำตอบของแต่ละอนุภาค (Evaluate) คือ การคำนวณค่าความเหมาะสม (Fitness value) ผ่านฟังก์ชันวัตถุประสงค์ (Objective Function) เพื่อนำไปใช้ในการปรับปรุงค่า PBEST และ GBEST ในขั้นตอนที่ 3 และ 4 ตามลำดับ

ขั้นตอนที่ 3

การปรับปรุงค่า PBEST (Update PBEST) คือ การที่แต่ละอนุภาคจะทำการเปรียบเทียบค่าความเหมาะสมของอนุภาคกับค่า PBEST ของตัวเอง หากดีกว่าจะทำการปรับปรุงค่า PBEST ตามค่าความเหมาะสมของอนุภาคที่ดีกว่า

ขั้นตอนที่ 4

การปรับปรุงค่า GBEST (Update GBEST) คือ การที่แต่ละอนุภาคจะทำการเปรียบเทียบค่าความเหมาะสมของอนุภาคกับค่า GBEST ของกลุ่ม หากดีกว่าจะทำการปรับปรุงค่า GBEST ตามค่าความเหมาะสมของอนุภาคที่ดีกว่า

ขั้นตอนที่ 5

การปรับปรุงความเร็ว (Update Velocity) คือ การปรับปรุงความเร็วของแต่ละอนุภาคเพื่อใช้ในการเคลื่อนที่ โดยนำค่า PBEST, GBEST และค่าความเร็วเดิมมาใช้ในการคำนวณ ดังสมการที่ 2.2

$$\vec{v}_k^{t+1} = \omega \vec{v}_k^t + c_1 r_1 (pbest - \vec{x}_k^t) + c_2 r_2 (gbest - \vec{x}_k^t) \quad (2.2)$$

โดยกำหนดให้

\vec{v}_k^{t+1} : ความเร็วในรอบปัจจุบันของอนุภาค

\vec{v}_k^t : ความเร็วเดิมของอนุภาค

$pbest$: ตำแหน่งที่ดีที่สุดที่อนุภาคตัวนั้นเคยพบเจอ

$gbest$: ตำแหน่งที่ดีที่สุดที่อนุภาคทุกตัวเคยพบเจอ

c_1, c_2 : ค่าคงที่ในการเรียนรู้

ω : ค่าสัมประสิทธิ์ความเฉื่อย

เอกสารนี้เป็นเอกสารที่... r_1, r_2 : ค่าที่ได้จากการสุ่มในช่วง $[0, 1]$ ยานเท่านั้น ไม่อนุภาคให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น \vec{x}_k^t : ตำแหน่งของอนุภาค และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 6

การปรับปรุงตำแหน่ง (Update Position) คือ การปรับปรุงตำแหน่งของอนุภาคให้เคลื่อนที่จากตำแหน่งเดิมไปยังตำแหน่งใหม่ โดยนำค่าที่ได้จากการปรับปรุงความเร็วในขั้นตอนที่ 5 มาใช้ในการคำนวณตามสมการข้างล่าง

$$\vec{x}_k^{t+1} = \vec{x}_k^t + v_k^{t+1} \quad (2.3)$$

โดยกำหนดให้

- \vec{x}_k^t : ตำแหน่งปัจจุบันของอนุภาค
- \vec{x}_k^{t+1} : ตำแหน่งใหม่ของอนุภาค
- v_k^{t+1} : ความเร็วในรอบปัจจุบันของอนุภาค

ขั้นตอนที่ 7

การตรวจสอบค่าคำตอบหรือค่าความเหมาะสม (Condition Satisfied) ขั้นตอนนี้เป็นการตรวจสอบคำตอบและกำหนดเงื่อนไขในการหยุดค้นหาคำตอบ คือถ้าเจอคำตอบหรือทำงานครบรอบการค้นหาที่กำหนดแล้วให้หยุดการวนซ้ำการทำงาน แต่ถ้าไม่เจอให้วนซ้ำการทำงานโดยกลับไปทำงานในขั้นตอนที่ 2 ใหม่อีกครั้ง

2.4 รูปแบบในการค้นหาคำตอบ

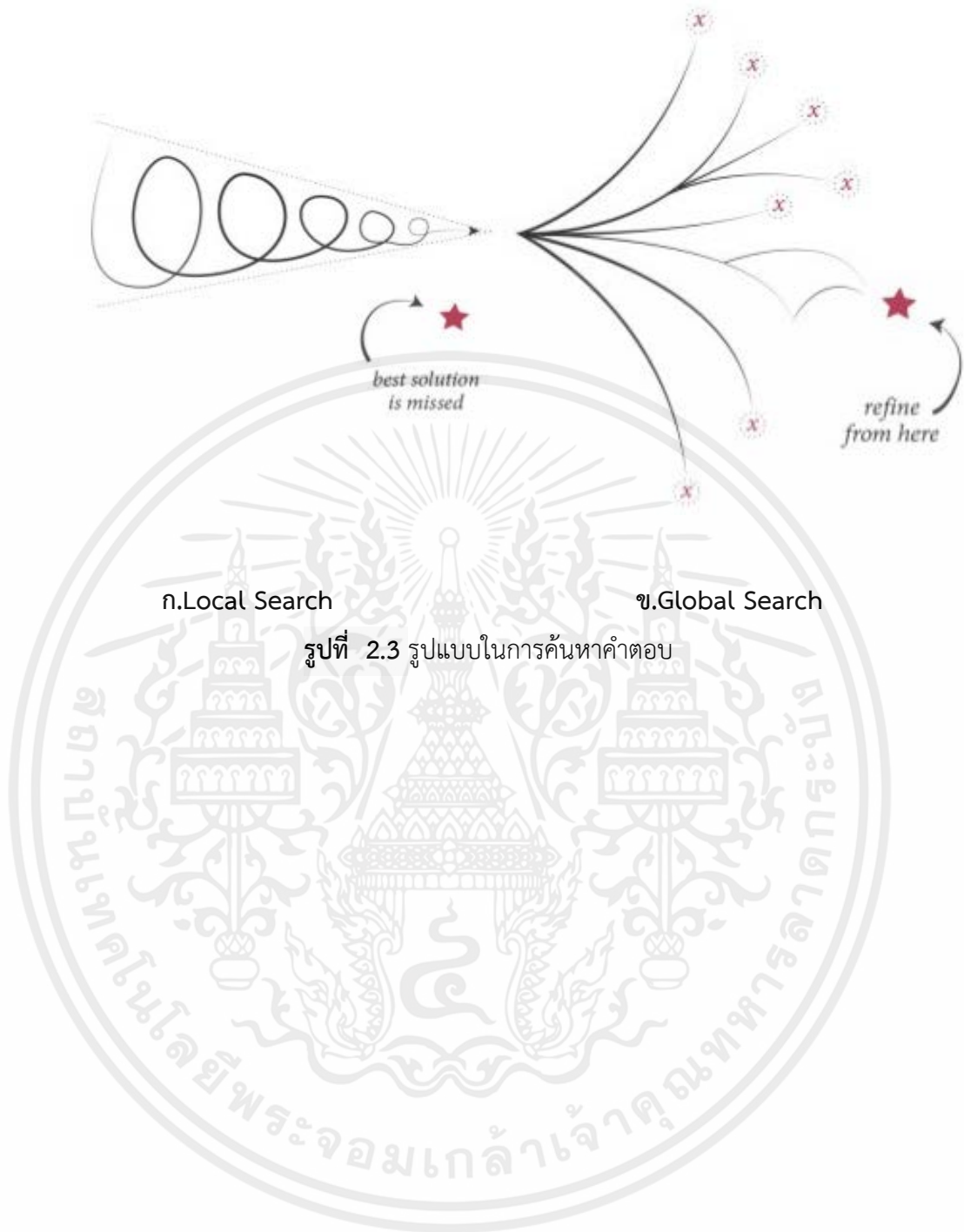
ลักษณะของการค้นหาคำตอบของ PSO มีอยู่ 2 รูปแบบ คือ

1. การค้นหาแบบ Local Search

การค้นหาแบบ Local Search เป็นการค้นหาใกล้ๆบริเวณตำแหน่งเดิม เพื่อทำการค้นหาตำแหน่งที่ดีกว่า แสดงดังรูป 2.3 ก.

2. การค้นหาแบบ Global Search

การค้นหาแบบ Global Search เป็นการค้นหาในแนวกว้าง ทำการค้นหาให้ไกลจากบริเวณตำแหน่งเดิม เพื่อหลีกเลี่ยงการติดใน Local Opima แสดงดังรูป 2.3 ข.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

งานวิจัยที่เกี่ยวข้อง

3.1 วิธีการค้นหาค่าความเหมาะสมแบบหลายกลุ่มอนุภาค (Multi Particle Swarm Optimization : MPSO)

วิธีการค้นหาค่าความเหมาะสมแบบหลายกลุ่มอนุภาค (Multi Particle Swarm Optimization) หรือ MPSO [3,14] คือการแบ่งกลุ่มอนุภาคของ PSO ออกเป็นหลายกลุ่มย่อย แต่ละกลุ่มย่อยจะให้ความสำคัญไปที่ขอบเขตที่จำเพาะ (specific region) ในขณะที่วิธีการกระจายแบบจำเพาะ (a specific diversification method) จะเป็นตัวตัดสินใจว่าแต่ละกลุ่มย่อยจะเริ่มกระบวนการทำงานที่ไหน หรือเมื่อไหร่ ซึ่งเป็นการเพิ่มความหลากหลายให้กับอนุภาค และปรับปรุงประสิทธิภาพในการค้นหาตำแหน่งใหม่

จากการทดลองของนักวิจัยที่หลากหลายพบว่า MPSO เหมาะสำหรับการค้นหาค่าความเหมาะสมของปัญหา โดยเฉพาะแบบ Multi-modal ที่มี Local optimum หลายจุด ซึ่งในปัญหาแบบ Multi-modal นี้สิ่งสำคัญก็คือการปรับสมดุลระหว่างการค้นหาแบบ exploration และ exploitation ดังนั้น MPSO จึงได้นำวิธีการในการปรับสมดุลนี้มาใช้ด้วยการแบ่งกลุ่มไปยังช่วงที่แตกต่างกัน แต่ละช่วงจะเน้นไปที่การค้นหาแบบ exploitation (individual sub-swarms) หรือแบบ exploration (diversification method) อย่างใดอย่างหนึ่งแทนที่จะพยายามเน้นการค้นหาทั้งสองแบบพร้อมกันเพราะจะมีผลทำให้กระบวนการค้นหามีประสิทธิภาพแยกลง [15]

ทั้งนี้การทำงานร่วมกันของแต่ละกลุ่มย่อยมีอยู่หลากหลายรูปแบบขึ้นอยู่กับวิธีการกระจายตัวของอนุภาคที่ดำเนินการโดยแต่ละ Multi-swarm ดังที่จะกล่าวในหัวข้อถัดไป

3.2 วิธีการค้นหาค่าความเหมาะสมแบบอนุภาคกลุ่มอย่างรวดเร็ว (Fast Particle Swarm Optimization : FPSO) และวิธีการค้นหาอย่างรวดเร็วแบบหลากหลายกลุ่มโดยการกลายพันธุ์และการผสมพันธุ์ (Fast Multi-swarm Optimization with Cauchy Mutation and Crossover operation : FMPSO)

วิธีการค้นหาค่าความเหมาะสมแบบอนุภาคกลุ่มอย่างรวดเร็ว (Fast Particle Swarm Optimization) หรือ FPSO [5] ถูกคิดค้นโดย Li, Zhang Liu และ Kang เพื่อแก้ปัญหาการติดใน

Local Optimum ของ PSO จึงได้เสนอให้มีการนำวิธีการกลายพันธุ์ (Mutation) เข้ามาช่วยในการกระจายตัวของอนุภาคให้กว้างขึ้น โดยวิธีการกลายพันธุ์ที่ FPSO เลือกใช้คือวิธีการกลายพันธุ์แบบคوشي (Cauchy Mutation) โดยได้นำมาประยุกต์ใช้กับความเร็วและตำแหน่งของอนุภาค ดังสมการข้างล่าง

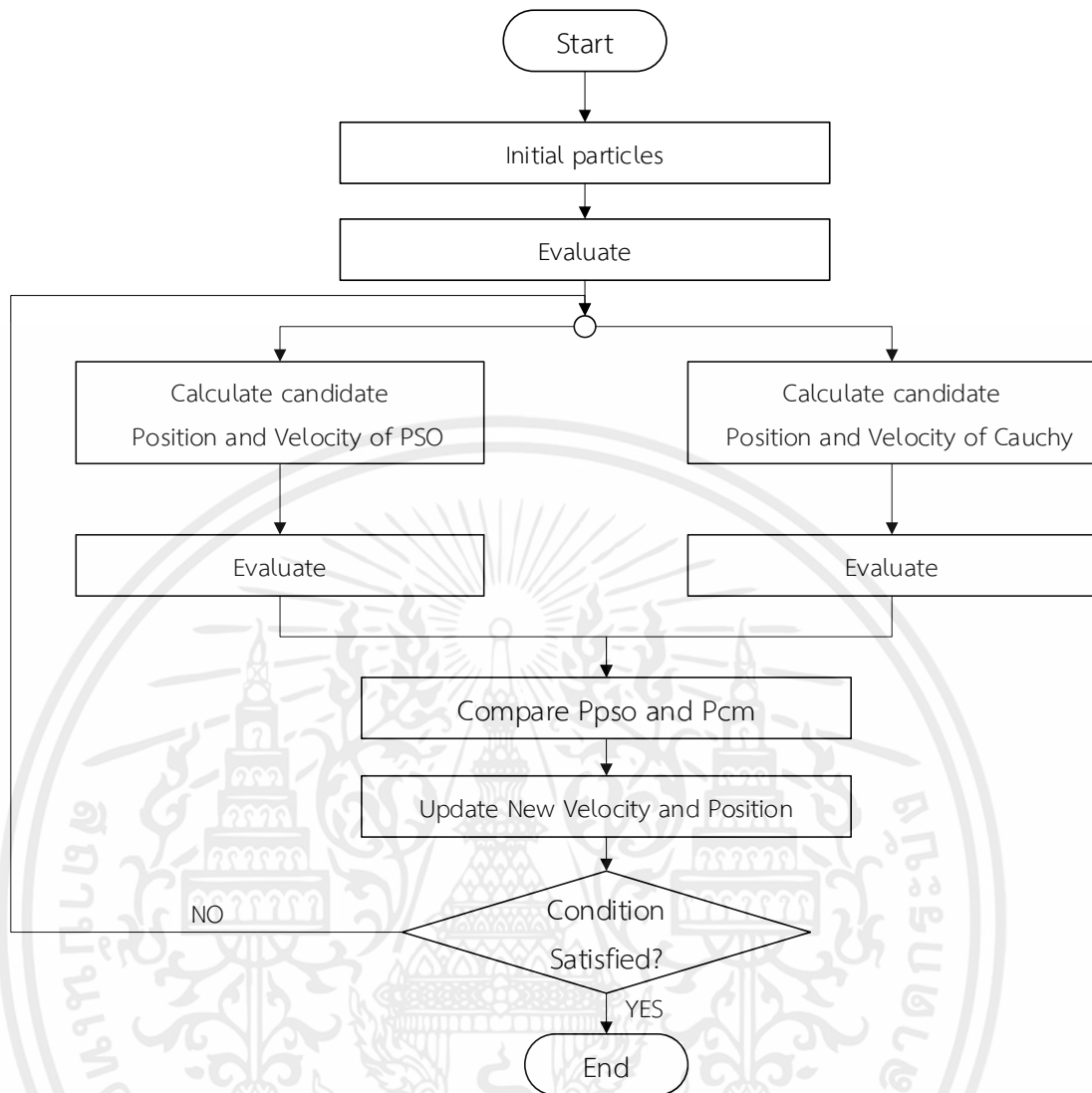
$$\vec{v}'_k = \vec{v}_k \exp(\delta) \quad (3.1)$$

$$\vec{x}'_k = \vec{x}_k + \vec{v}'_k \delta_k \quad (3.2)$$

โดยที่ δ และ δ_k คือ ค่าที่ได้จากสุ่มของคوشي

การทำงานของ FPSO ได้เพิ่มขั้นตอนในการคำนวณตำแหน่งและความเร็วแบบคوشي โดยทำควบคู่กับการทำงานแบบ PSO ในส่วนของการคำนวณตำแหน่งและความเร็วสามารถอธิบายการทำงานได้จากฝั่งที่ (3.1) ในขั้นตอนแรกจะทำการสุ่มประชากรเริ่มต้นก่อนและทำการประมวลผลของอนุภาคแต่ละตัว จากนั้นทำการคำนวณตำแหน่งและความเร็วแบบ PSO ตามสมการที่ (2.2) และ (2.3) ควบคู่ไปกับการคำนวณตำแหน่งและความเร็วแบบคوشيตามสมการที่ (3.1) และ (3.2) จากนั้นนำค่าที่ได้จากการคำนวณของแต่ละวิธีมาประมวลผลเพื่อนำค่าความเหมาะสมที่ได้จาก PSO (P_{PSO}) มาเปรียบเทียบกับค่าความเหมาะสมที่ได้จาก Cauchy Mutation (P_{CM}) หากค่าไหนดีกว่าจะนำค่าตำแหน่งและความเร็วที่ได้จากวิธีนั้นมาใช้ในการปรับปรุงตำแหน่งและความเร็วนั่นเอง

การใช้เทคนิคคوشيมาช่วยทำให้ FPSO สามารถหลีกเลี่ยงการติดใน Local Optimum ได้และมีโอกาสไปค้นหาคำตอบในบริเวณอื่นเพื่อเจอค่าตำแหน่งที่ดีกว่านั่นเอง แต่ FPSO ก็ยังคงมีปัญหาในการที่ลู่เข้าหาคำตอบและติดอยู่ใน Local Optimum ที่เร็วเกินไป จึงได้นำแนวคิดการแบ่งกลุ่มอนุภาคออกเป็นหลายกลุ่มมาช่วยในการเพิ่มความสามารถในการค้นหาคำตอบให้กว้างขึ้นและเพิ่มโอกาสในการเจอคำตอบได้เร็วขึ้น Li, Zhang Liu และ Kang จึงได้พัฒนา FPSO เป็น FMPSO [16]



รูปที่ 3.1 แผนผังการทำงานของ FPSO

วิธีการค้นหาอย่างรวดเร็วแบบหลากหลายกลุ่มโดยการกลายพันธุ์และการผสมพันธุ์ (Fast Multi-swarm Optimization with Cauchy Mutation and Crossover operation) หรือ FMPSO เป็นอัลกอริทึมที่พัฒนามาจาก FPSO โดย FMPSO ทำการแบ่งอนุภาคเป็นกลุ่มย่อยๆ หลายกลุ่ม (multi-swam) และนำขั้นตอนการ Crossover มาใช้เพื่อเพิ่มประสิทธิภาพในการลู่เข้าของอนุภาค ในแต่ละรอบการทำงานของอนุภาคอนุภาคจะทำการปรับปรุงตามค่าที่ดีที่สุดของกลุ่มตัวเอง

และปรับปรุงตามค่าที่ดีที่สุดของกลุ่มอื่น ข้อมูลจะถูกใช้ร่วมกันและเรียนรู้จากกลไกการ Crossover ที่ทำให้ฝูงนกขยายพื้นที่ในการค้นหาซึ่งทำให้ความเร็วในการลู่เข้าของอนุภาคมีประสิทธิภาพมากขึ้น

ทั้งนี้การทำงานของ FMPSO แสดงให้เห็นตามผังงานที่ (3.2) และสามารถอธิบายขั้นตอนการ Crossover ของ FMPSO ได้ดังนี้

ขั้นตอนแรก: สำหรับแต่ละอนุภาคของกลุ่ม k , ให้ทำการสุ่มเลือกอนุภาค P_b ที่ดีที่สุดจากกลุ่มที่ถูกสุ่มได้

ขั้นตอนที่สอง: สำหรับแต่ละมิติ i ของอนุภาค P_b โดยกำหนดให้ตำแหน่งและความเร็วของอนุภาค P ที่มีมิติ i คือ $P_x[i]$ และ $P_v[i]$ ตามลำดับ ให้ทำการสุ่มค่าอนุภาคมา 1 ค่าถ้าค่าที่ได้จากการสุ่มน้อยกว่าอัตราการ Crossover (qc) ให้ทำการ Crossover ระหว่าง อนุภาค P กับ P_b ตามสมการต่อไปนี้

$$P_x[i] = (1 - \alpha) * P_x[i] + \alpha * P_{bx}[i] \quad (3.3)$$

$$P_v[i] = rand() * (P_{bx}[i] - P_x[i]) \quad (3.4)$$

โดยที่ α คือ ค่าที่ได้จากการสุ่มในช่วง 0 - 1

i คือ มิติของอนุภาค P

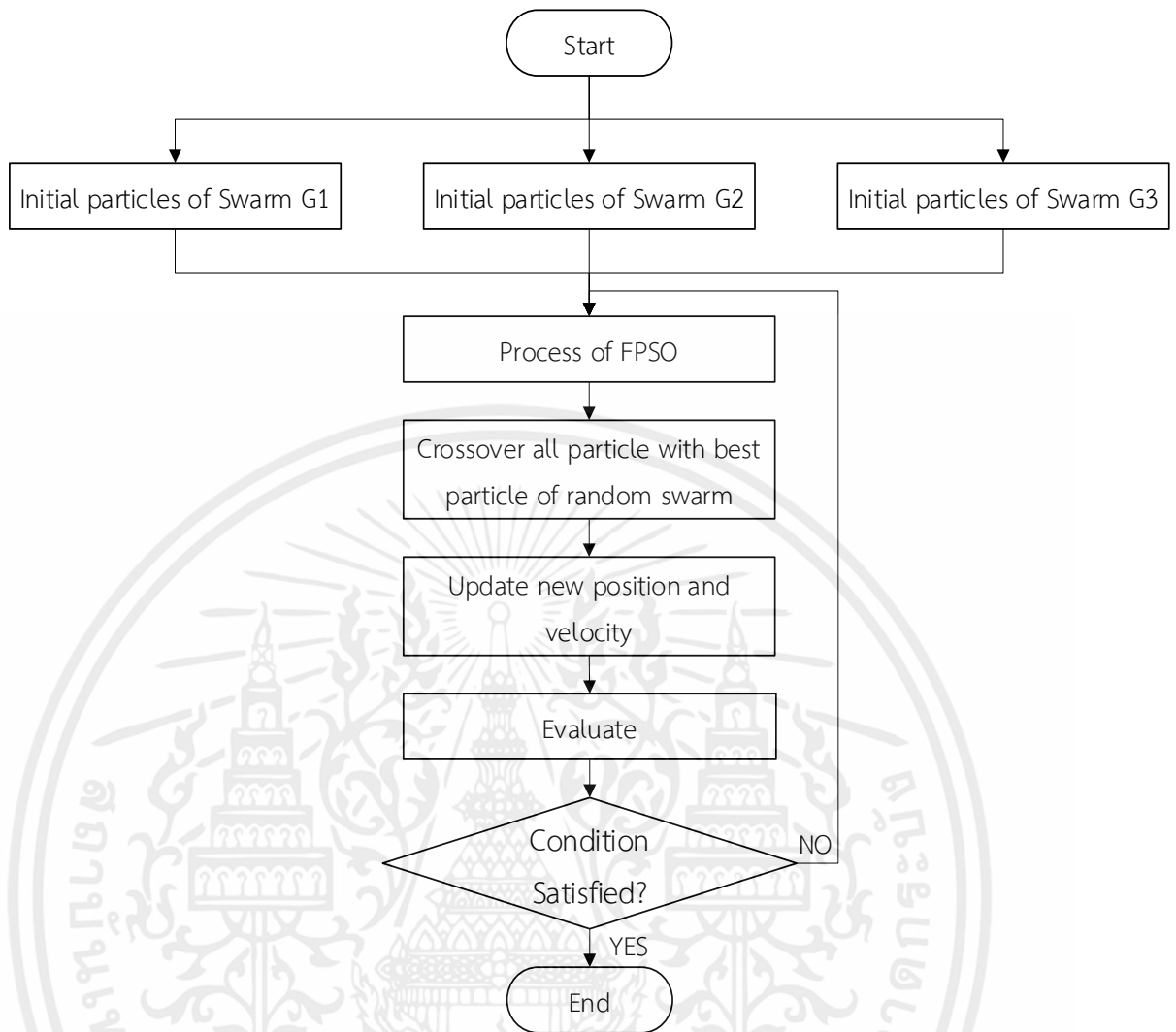
$P_x[i]$ คือ ตำแหน่งของอนุภาค P มิติที่ i

$P_v[i]$ คือ ความเร็วของอนุภาค P มิติที่ i

P_{bx} คือ อนุภาคที่ดีที่สุด

ขั้นตอนที่สาม: ถ้าทุกอนุภาคของกลุ่มอนุภาค k ได้รับการปรับปรุงค่าตำแหน่งและความเร็วครบหมดแล้วให้หยุดการทำงาน แต่ถ้ายังปรับปรุงไม่หมดหรือไม่เข้าเงื่อนไขในการปรับปรุงให้กลับไปทำงานที่ขั้นตอนที่ 1 อีกครั้ง

วิธี FMPSO นี้แม้จะทำให้ได้ผลลัพธ์ในการค้นหาคำตอบดีขึ้น แต่ก็เพิ่มจำนวนการเดาคำตอบในแต่ละรอบของการค้นหาเช่นกัน แม้ว่าจะเพิ่มขั้นตอนของการ Crossover มาช่วยในการลดขั้นตอนเอกสารนี้เป็นการเดาคำตอบแล้วก็ตามับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนผังการทำงานของ FMPSO

3.3 ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบหลากหลายกลุ่มโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ร่วมกับการเปลี่ยนตำแหน่งของอนุภาคที่ดีที่สุดของกลุ่ม (Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm : MPSOSM)

เอกสารนี้เป็นเอกสาร MPSOSM [7] ถูกพัฒนามาเพื่อแก้ปัญหาการติด Local Optimum ของ PSO โดยวิธีนี้ผู้วิจัยได้นำแนวคิดการค้นหาแบบ Multi-Swarm มาช่วยเพิ่มความสามารถในการค้นหาคำตอบแบบ Global ร่วมกับการ Reposition เมื่ออนุภาคของแต่ละกลุ่มติดอยู่ใน Local Optimum รวมถึงมีการ

Reposition ตำแหน่งของ GBEST เมื่อทุกกลุ่มติดใน Local Optimum ที่จุดเดียวกัน โดยฟังก์ชันที่ (3.3) แสดงถึงวิธีการทำงานของ MPSOSM

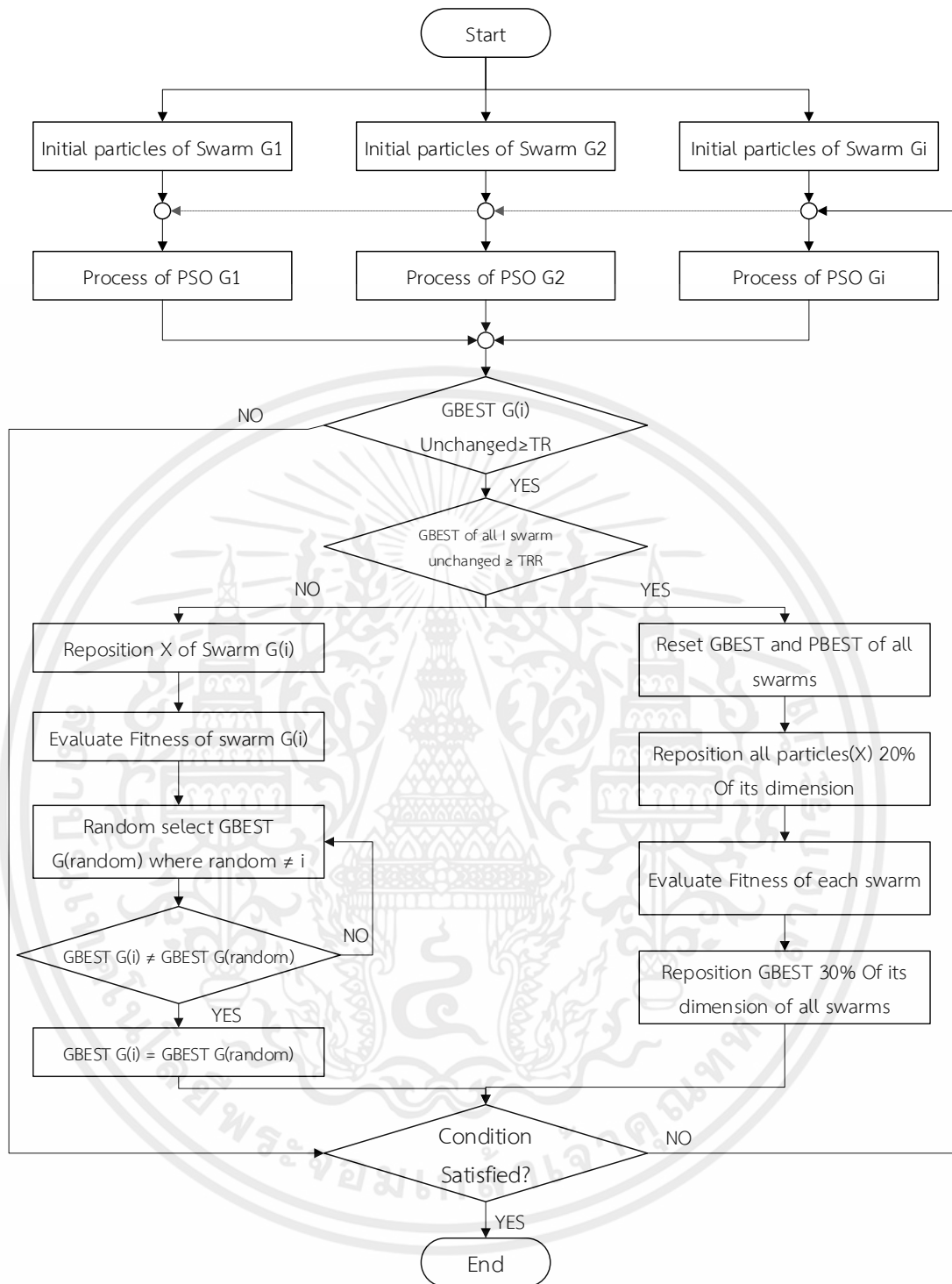
จากฟังก์ชันที่ (3.3) สามารถอธิบายได้ว่า MPSOSM เริ่มต้นทำงานแบบหลายกลุ่มอนุภาค โดยแต่ละกลุ่มย่อยจะทำการปรับปรุงตำแหน่งและความเร็วตามแนวคิดของ PSO ต้นแบบจนเจอคำตอบหรือจนกระทั่งมีกลุ่มใดกลุ่มหนึ่งติด Local optimum ให้ทำการ Reposition อนุภาคตามสมการที่ (3.5) และใช้ค่า GBEST ใหม่เป็นค่า GBEST ของกลุ่มอื่นที่ได้จากการสุ่ม ซึ่งขั้นตอนนี้เป็น การเพิ่มโอกาสของฝูงเพื่อหลีกเลี่ยงการติด Local optimum ด้วยการเคลื่อนที่ไปยัง GBEST ของกลุ่มอื่นแทน แต่ถ้าหากทุกกลุ่มย่อยติด Local Optimum ที่จุดเดียวกัน ทุกกลุ่มจะทำการ Reset PBEST และ GBEST ของทุกกลุ่มจากนั้นทำการ Reposition ตำแหน่งของอนุภาคตามสมการที่ (3.5) ทำการคำนวณค่าความเหมาะสมของอนุภาคทุกตัวรวมทั้งปรับเปลี่ยนค่า PBEST และ GBEST ที่ถูก Reset ไว้ก่อนหน้า แล้วจึง Reposition ค่าตำแหน่ง GBEST ของทุกกลุ่มตามสมการที่ (3.6) เพื่อให้ GBEST ห่างจากค่า GBEST เดิมตอนที่ติด Local Optimum เพื่อไม่ให้อนุภาคเคลื่อนที่กลับมายังจุดที่ติด Local Optimum จุดเดิม โดยสมการที่ใช้ในการ Reposition นั้นจะเป็นการเลือกเปลี่ยนตำแหน่งของอนุภาคเพียงเล็กน้อย เพื่อให้อนุภาคนั้นยังคงค้นหาคำตอบไม่ไกลจากบริเวณเดิม ซึ่งเป็นการช่วยลดเวลาในการค้นหาและลู่เข้าหาคำตอบ

$$X_{id} = \pm(X_{id} \times (0.85 + 0.3 \times rand())) \quad (3.5)$$

$$GBEST_{id} = \pm(GBEST_{id} \times (0.85 + 0.3 \times rand())) \quad (3.6)$$

MPSOSM นี้สามารถแก้ปัญหาการติดใน Local Optimum ได้ และเจอคำตอบเกือบทุกสมการการค้นหา แต่ยังคงใช้เวลาในการลู่เข้าหาคำตอบที่นานขึ้นเนื่องจากมีจำนวนอนุภาคต่อกลุ่มน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

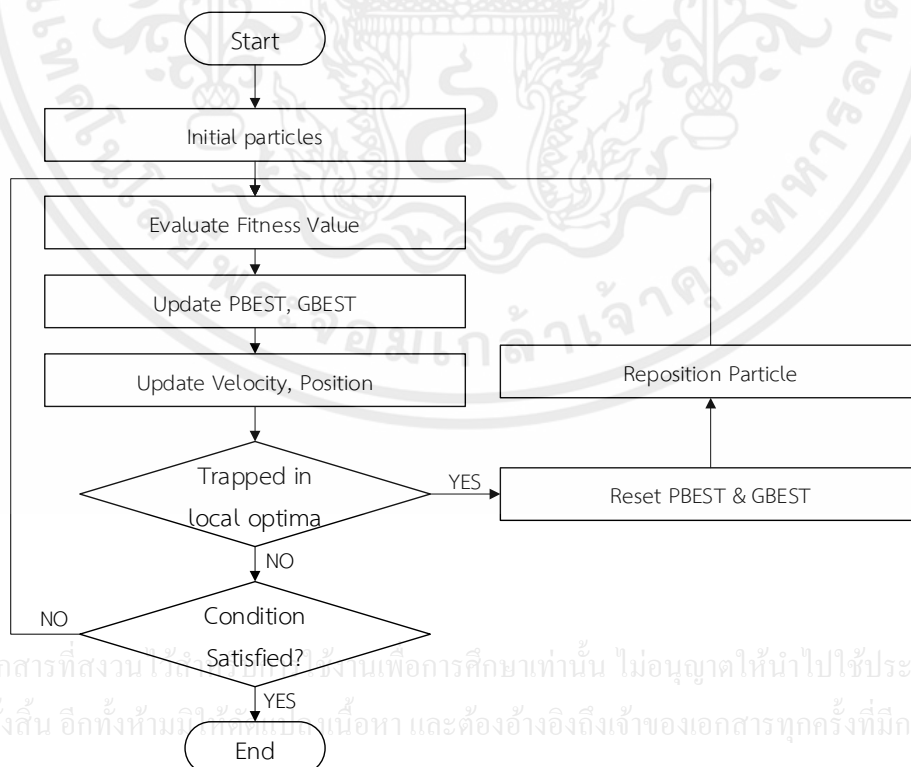


รูปที่ 3.3 แผนผังการทำงานของ MPSOSM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ (Reposition by Slightly Mutation Particle with Particle Swarm Optimization : RPSO)

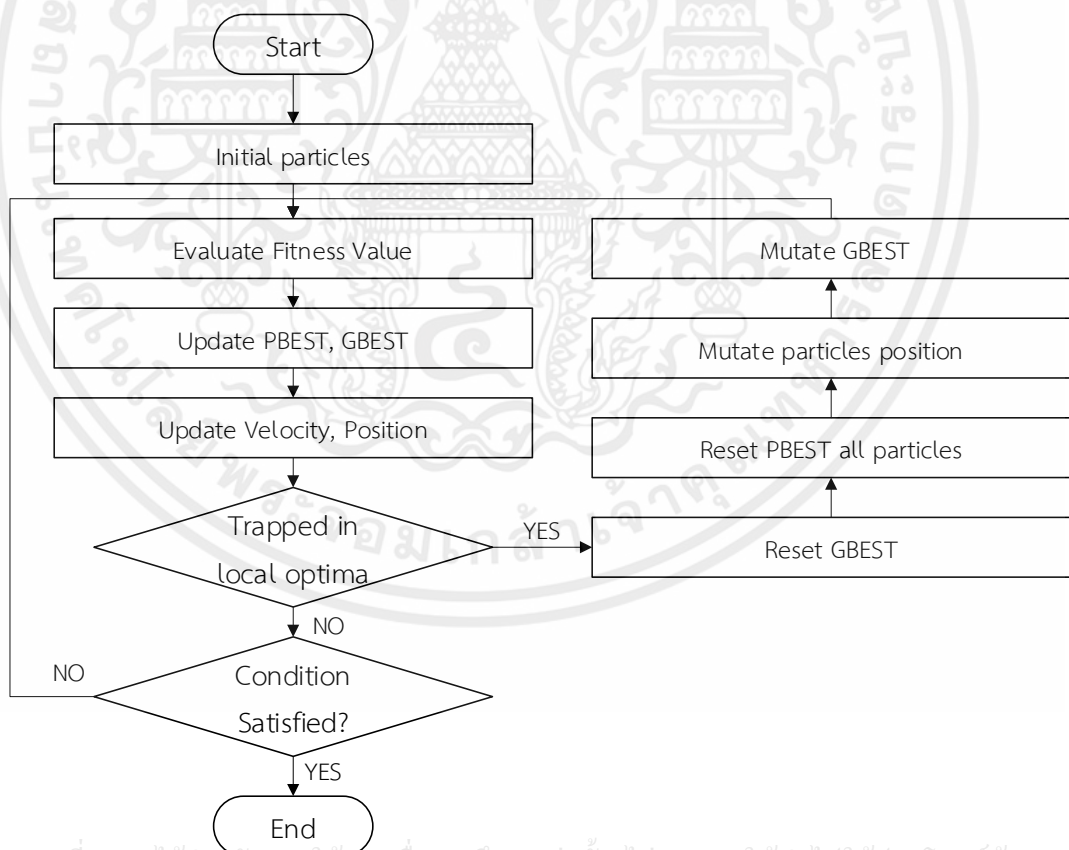
RPSO เป็นการค้นหาแบบหนึ่งกลุ่มอนุภาค เพื่อรักษาความสามารถในการค้นหาคำตอบแบบ Local Search โดยมีการทำงานในตอนเริ่มเช่นเดียวกับ PSO ปกติแต่ในตอนที่อนุภาคติดอยู่ใน Local Optimum จะนำเทคนิคการกระจายตำแหน่งของอนุภาคมาช่วยในการหลีกเลี่ยงการติดในจุด Local Optimum พร้อมทั้งมีการ Reset GBEST เพื่อไม่ให้อนุภาคกลับไปติด Local Optimum เดิม โดย RPSO จะเริ่มต้นทำงานแบบ PSO ด้วยการสุ่มประชากรเริ่มต้น เมื่อได้ตำแหน่งเริ่มต้นแล้วทำการ Evaluate เพื่อหาค่าความเหมาะสมพร้อมทั้งปรับปรุงค่า PBEST และ GBEST จากนั้นนำค่าที่ได้มาใช้ในการปรับปรุงทิศทางและความเร็วในการเคลื่อนที่ของแต่ละอนุภาค จะวนทำกระบวนการข้างต้นไปเรื่อย ๆ จนเจอคำตอบ หรือจนติดในจุด Local Optimum จะเข้าสู่กระบวนการ RPSO คือ จะทำการ Reset PBEST และ GBEST จากนั้นจะทำการ Reposition เพื่อกระจายอนุภาคให้หลุดจาก Local optimum และเพิ่มโอกาสในการค้นหาคำตอบให้มากขึ้น ดังจะเห็นได้จากแผนผังการทำงาน RPSO ที่ (3.4)



รูปที่ 3.4 แผนผังการทำงานของ RPSO

3.5 วิธีการเริ่มต้นการทำงานใหม่ของการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาคด้วยการกลายพันธุ์ของตำแหน่งที่ดีที่สุดของอนุภาค (The use of Global Best position in rerun of particle swarm optimization : GBPSO)

GBPSO[8] ถูกพัฒนามาเพื่อแก้ปัญหาการติดใน Local Optimum ของ PSO โดยอัลกอริทึมนี้จะเริ่มต้นทำงานเหมือน PSO ต้นแบบ จนเมื่ออนุภาคติดอยู่ใน Local Optimum ตามรอบที่กำหนดจะเข้าสู่กระบวนการของ GBPSO คือทำการกระจายตำแหน่งของอนุภาค PBEST และตำแหน่งของอนุภาคที่ดีที่สุด GBEST ด้วยการเลือกเปลี่ยนตำแหน่งของอนุภาคและ GBEST เพียงเล็กน้อยเพื่อให้อนุภาคนั้นยังคงค้นหาค่าตอบไม่ไกลจากบริเวณเดิม ซึ่งเป็นการช่วยลดเวลาในการค้นหาและหลุดเข้าหาค่าตอบ จากนั้นจะทำการเริ่มต้นการทำงานของ PSO อีกครั้ง จนกว่าจะติดใน Local Optimum ครั้งถัดไป หรือเจอค่าคำตอบที่ดีที่สุดถึงจะหยุดกระบวนการทำงาน ดังจะเห็นได้จากรูปที่ 3.5



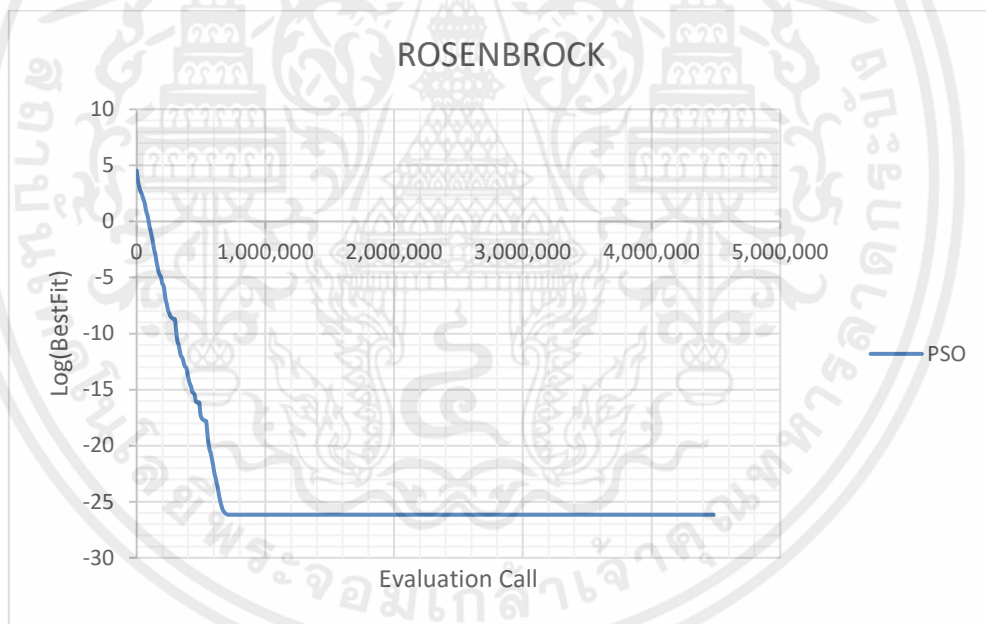
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 แผนผังการทำงานของ GBPSO

บทที่ 4

งานวิจัยที่นำเสนอ

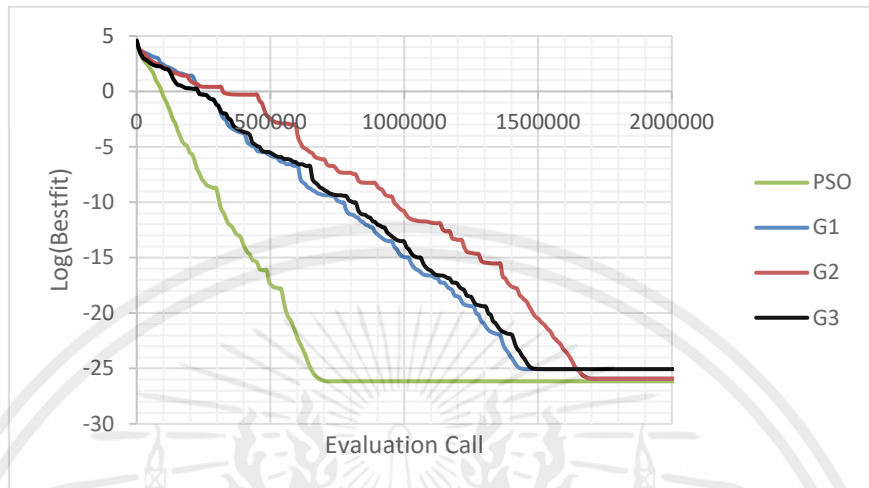
เพื่อแก้ปัญหาการติดใน Local Optimum ของ PSO ที่ทำให้ PSO ไม่สามารถหาค่าคำตอบที่ดีที่สุดได้ ดังจะเห็นได้จากรูปที่ 4.1 ที่แสดงถึงพฤติกรรมการค้นหาคำตอบของ PSO จากกราฟจะเห็นว่าในช่วงแรกการค้นหาคำตอบของ PSO อนุภาคมีการลู่เข้าหาค่าตอบอย่างต่อเนื่องระยะเวลาหนึ่งจนมาถึงในช่วงที่กราฟคงที่ นั่นแสดงให้เห็นว่า PSO ไม่สามารถหาค่าคำตอบที่ดีกว่าเดิมได้หรือติดอยู่ใน Local Optimum นั่นเอง ซึ่งเป็นปัญหาสำคัญของ PSO จากปัญหานี้ นักวิจัยหลายท่านจึงได้ทำการปรับปรุงพฤติกรรมการค้นหาคำตอบของ PSO เพื่อหลีกเลี่ยงปัญหาเหล่านี้ ไม่ว่าจะเป็นการกลายพันธุ์ การรีโพลซิชัน การแบ่งกลุ่มอนุภาคออกเป็นกลุ่มย่อย และวิธีอื่น ๆ อีกมาก



รูปที่ 4.1 กราฟ PSO แสดงพฤติกรรมการค้นหาคำตอบของฟังก์ชัน ROSENBROCK

การแบ่งกลุ่มอนุภาคออกเป็นกลุ่มย่อย หรือ MPSO ถือเป็นอีกทางเลือกในการแก้ปัญหาดังกล่าวที่เป็นที่นิยม เนื่องจากเป็นการเพิ่มโอกาสในการค้นหาคำตอบได้กว้างขึ้นนั่นเอง ในขณะที่เดียวกันก็เป็นการลดความสามารถในการค้นหาเฉพาะที่ลงด้วย เนื่องจากจำนวนอนุภาคต่อกลุ่มก็จะลดลงไปด้วยเช่นกัน มีผลให้ MPSO ลู่เข้าหาค่าตอบและติด Local Optimum ได้ช้ากว่า PSO ดังจะเห็นได้จากกราฟดังรูปที่ 4.2 โดยในกราฟ PSO จะมีประชากร 90 อนุภาค และมีเพียง 1 กลุ่ม ส่วน MPSO

มี 3 กลุ่ม (G1 G2 G3) กลุ่มละ 30 อนุภาค เนื่องจากมีจำนวนอนุภาคน้อยกว่าจึงทำให้กราฟลู่ลงช้ากว่า และจะเห็นว่าตำแหน่งที่ G1 G2 และ G3 ติด Local Optimum ที่คนละตำแหน่งกัน

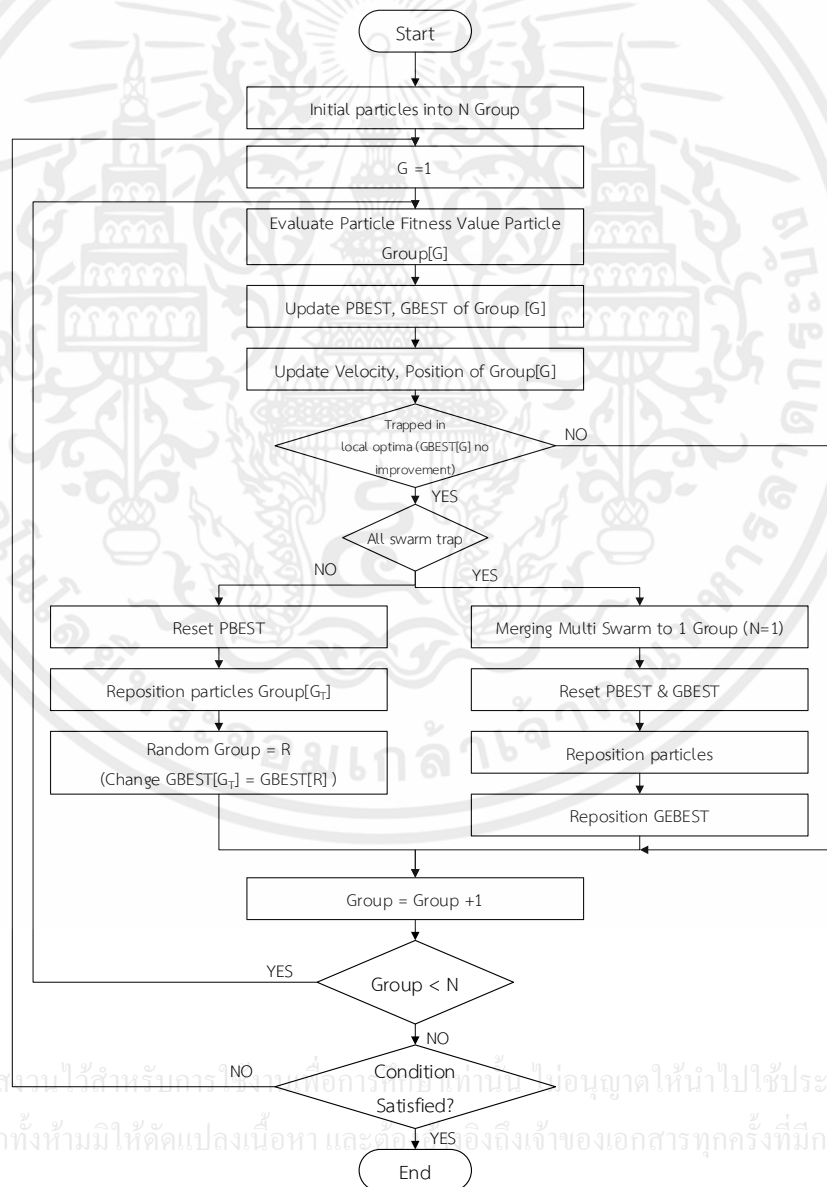


รูปที่ 4.2 กราฟเปรียบเทียบการติด Local Optimum ระหว่าง PSO กับ MPSO ในการค้นหาคำตอบของฟังก์ชัน ROSEN BROCK

อัลกอริทึมแบบ MPSO ที่แสดงในรูป 4.2 แต่ละกลุ่มมีการค้นหาที่เป็นอิสระต่อกันทำให้เกิดการค้นหาที่กว้างกว่าโดยจะเห็นว่าทั้ง 3 กลุ่มติดอยู่ที่จุด Local Optimum คนละตำแหน่งกัน แต่อัลกอริทึมในกลุ่มนี้จะต้องมีการสื่อสารข้อมูลกันระหว่างกลุ่มเพื่อให้แต่ละกลุ่มสามารถนำข้อมูลของกลุ่มอื่นมาใช้ในการค้นหาต่อไปได้ โดยอัลกอริทึมที่เลือกมาศึกษาและพัฒนาต่อคือ ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบหลากหลายกลุ่มโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ร่วมกับการเปลี่ยนตำแหน่งของอนุภาคที่ดีที่สุดของกลุ่ม หรือ MPSOSM [7]

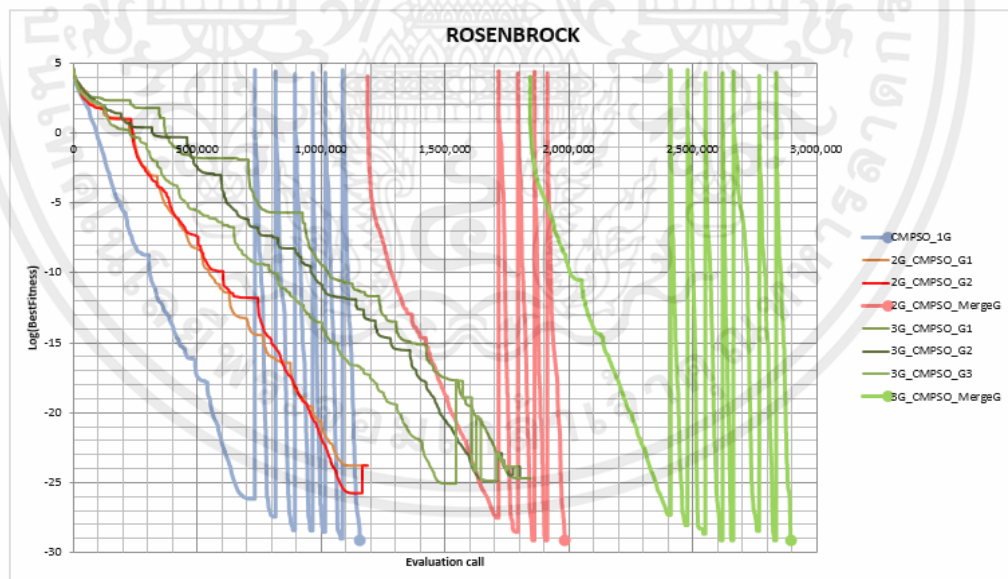
จาก MPSOSM จะเห็นว่า MPSOSM ใช้วิธีแก้ปัญหาการติดใน Local Optimum ของ PSO โดยการแบ่ง PSO ออกเป็นหลายกลุ่มแต่ละกลุ่มมี GBEST ของตัวเอง ดังนั้นแต่ละกลุ่มจะมีการเคลื่อนที่ของกลุ่มที่เป็นอิสระต่อกันจนกระทั่งกลุ่มใดกลุ่มหนึ่งติดอยู่ใน Local Optimum จะใช้การกลายพันธุ์ตำแหน่งของอนุภาค รวมถึงการใช้ตำแหน่ง GBEST จากกลุ่มอื่นเพื่อหลีกเลี่ยง Local Optimum แต่การแบ่งอนุภาคออกเป็นหลายกลุ่มทำให้จำนวนอนุภาคต่อกลุ่มลดลงมีผลให้การหาค่าตอบช้าลง นอกจากนี้ในที่สุดทุกกลุ่มก็ติดอยู่ใน Local Optimum เดียวกัน แต่ละกลุ่มจึงใช้การกลายพันธุ์ของอนุภาคของกลุ่มตัวเอง และแต่ละกลุ่มกลายพันธุ์ GBEST จาก GBEST ค่าเดียวกัน โดย

แต่ละกลุ่มกลายพันธุ์อิสระต่อกันและยังคงมี GBEST ของตัวเอง จึงมีการค้นหาที่เป็นอิสระต่อกัน เนื่องจากการแบ่งออกเป็นหลายกลุ่มทำให้การค้นหากว้างขึ้นแต่ก็มีข้อเสียที่ใช้เวลานานในการลู่เข้าหา Local Optimum ตามที่ได้กล่าวมาแล้ว ทางผู้วิจัยจึงได้ทำการพัฒนาและปรับปรุงด้วยการลดจำนวนกลุ่มลงในช่วงเวลาที่เหมาะสมซึ่งเป็นการเพิ่มความสามารถในการค้นหาค่าตอบเฉพาะที่แบบ Local Search เพื่อลดเวลาการค้นหาค่าตอบของ MPSOSM ลง พร้อมกับปรับปรุงสมการในการ Reposition ให้อยู่ในช่วงที่แคบกว่าเดิม โดยใช้ชื่อวิธีที่ทำการปรับปรุงและพัฒนาคือ **ขั้นตอนวิธีการรวมกลุ่มอนุภาคแบบหลากหลายกลุ่มเพื่อหาค่าความเหมาะสม (Combine multi particle swarm in supporting trapping in local optima [9])** หรือ CMPSO โดยการทำงานของ CMPSO จะแสดงดังรูปที่ 4.3



รูปที่ 4.3 แผนผังการทำงานของ CMPSO

จากการทดลองในงานวิจัย CMPSO (ภาคผนวก ข) จะเห็นว่าเมื่อ CMPSO ทำการลดจำนวนกลุ่มลงในเวลาที่เหมาะสม คือ ตอนที่ทุกกลุ่มติดอยู่ที่ Local Optimum ตำแหน่งเดียวกัน เนื่องจากทุกกลุ่มค้นหาในตำแหน่งเดียวกันการเพิ่มจำนวนอนุภาคต่อกลุ่ม(รวมกลุ่ม)จึงมีโอกาสทำให้เพิ่มความสามารถในการค้นหาเฉพาะที่แบบ Local Search มีผลให้การลู่เข้าหาคำตอบมีประสิทธิภาพมากขึ้น ทางผู้วิจัยได้ทำการทดสอบโดยการแบ่งอนุภาคเป็น 1 กลุ่ม 2 กลุ่ม และ 3 กลุ่มในการเริ่มต้นการค้นหาหาคำตอบ จากนั้นทำการลดจำนวนกลุ่มอนุภาคลงเหลือ 1 กลุ่ม เมื่อทุกกลุ่มอนุภาคติดใน Local Optimum ที่จุดเดียวกัน ซึ่งจากผลการทดลองที่แสดงบนรูปที่ 4.4 จะเห็นว่าการเริ่มต้นค้นหาด้วยจำนวนกลุ่มที่น้อยกว่านั้นสามารถค้นหาคำตอบได้เร็วกว่า จึงสามารถสรุปได้ว่าจำนวนกลุ่ม (จำนวนอนุภาคในกลุ่ม)มีผลต่อการลู่เข้าหาคำตอบเนื่องจากเมื่อลดจำนวนกลุ่มที่ใช้ในการค้นหาจำนวนอนุภาคต่อกลุ่มอนุภาคก็เพิ่มขึ้นทำให้การลู่เข้าหาคำตอบได้เร็วขึ้นนั่นเอง ในกรณีนี้เป็นการค้นหาจุด Optimum ของฟังก์ชัน Rosenbrock 100 มิติ โดยทั้ง แบบ 1 กลุ่ม 90 อนุภาค 2 กลุ่ม กลุ่มละ 45 อนุภาค และ 3 กลุ่ม กลุ่มละ 30 อนุภาค ทุกวิธีเจอคำตอบที่จุด Optimum แต่วิธีที่มีจำนวนอนุภาคในกลุ่มมากกว่าเจอคำตอบก่อนตามลำดับ



รูปที่ 4.4 กราฟแสดงการทดสอบการแบ่งจำนวนกลุ่มอนุภาคเพื่อทดสอบ CMPSO

เอกสารนี้เป็นเอกสาร จากผลทดลองที่แสดงในรูปที่ 4.4 ทางผู้วิจัยจึงได้นำแนวคิดนี้มาปรับปรุงอัลกอริทึม CMPSO อีก ไม่ว่าจะเป็นกรณีใดก็ตาม โดยกำหนดให้ในช่วงแรกของการค้นหาคำตอบนั้นทำการค้นหาคำตอบด้วยกลุ่มอนุภาคเพียง 1 กลุ่ม แทนการค้นหาด้วย Multi Group แบบ CMPSO เพื่อให้อนุภาคทำการลู่เข้าหาคำตอบได้เร็วขึ้น

ในช่วงแรก ซึ่งอนุภาคปรับตำแหน่งและความเร็วตามแนวคิดของ PSO ต้นแบบไปจนเมื่อทุกอนุภาคติดใน Local Optimum จะทำการ Reposition ตำแหน่งของอนุภาคและ GBEST โดยถ้าติดอีกก็จะทำขั้นตอนนี้ต่อไปอีก 6 ครั้ง จากนั้นถ้าติด Local Optimum จะทำการแบ่งกลุ่มอนุภาคออกเป็นหลายๆกลุ่มย่อยเพื่อเพิ่มความสามารถในการค้นหาแบบ Global Search และทำการ Reposition ตำแหน่งของอนุภาคจากตำแหน่งเดิมตอนที่ติด Local Optimum และตำแหน่งของ GBEST ซึ่งจะมี GBEST ที่ถูก Reposition อย่างอิสระต่อกันเท่ากับจำนวนกลุ่ม โดยที่แต่ละกลุ่มจะแยกกันดำเนินการค้นหาค่าคำตอบอย่างอิสระจนแต่ละกลุ่มมีการติดใน Local Optimum อีกครั้ง ในกรณีที่มีเพียงแค่บางกลุ่มเท่านั้นที่ติดใน Local Optimum ให้ทำการ Reposition ตำแหน่งของอนุภาคและใช้ GBEST ของกลุ่มอื่นที่ได้จากการสุ่มมาเป็น GBEST ของกลุ่มที่ติด Local Optimum แต่ถ้าทุกกลุ่มมีการติด Local Optimum ที่จุดเดียวกันให้แต่ละกลุ่มทำการ Reposition ตำแหน่งของอนุภาคและตำแหน่งของ GBEST พร้อมทำการตรวจสอบรอบในการติด Local Optimum ถ้าครบตามรอบที่กำหนดไว้ (4 ครั้ง) จะทำการลดจำนวนกลุ่มย่อยเป็นกลุ่มเดียวกัน และกลับไปสู่ขั้นตอนในการค้นหาในรูปแบบ Single Group อีกครั้ง โดย MSPSO นี้จะทำการค้นหาสลับกันระหว่าง Single Group กับ Multi-Group ไปจนเจอคำตอบที่เหมาะสมหรือครบรอบการทำงานที่กำหนด ซึ่งเราเรียกวิธีข้างต้นว่า **วิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดในโลคอลออปติ멈ของพีเอสโอ (Merge and split particle swarm in reducing trapping in local optima of PSO, MSPSO)**

MSPSO มีขั้นตอนการทำงานสำคัญดังจะอธิบายต่อไปนี้

1. การรีเซ็ตค่า PBEST และ GBEST

กระบวนการรีเซ็ตค่า PBEST และ GBEST จะเกิดขึ้นทั้งใน Single Group และ Multi Group และเกิดขึ้นก่อนกระบวนการ Reposition โดยจะทำการรีเซ็ตค่า PBEST และ GBEST ไปที่ค่าสูงสุด (maximum) เพื่อป้องกันไม่ให้อนุภาคหลุดเข้าหาค่า PBEST และ GBEST เดิมที่ทำให้อนุภาคติดใน Local Optimum และจะได้มีการกำหนดค่า GBEST ใหม่ ซึ่งเป็นค่าของตำแหน่งที่ดีที่สุดของกลุ่มอนุภาคหลังจากการ Reposition โดยถ้าไม่มีการ Reset ค่า GBEST จะเป็นค่า GBEST ณ ตำแหน่งที่ติด Local Optimum ซึ่งความจริงจะมีการ Reposition GBEST อีกที ดังนั้นการที่ไม่ Reset GBEST ก็มีการย้ายตำแหน่ง GBEST จากการ Reposition อยู่ดีเพียงแต่การ Reset ค่า GBEST จะได้ GBEST

เอกสารนี้เป็นที่ไกลจากจุด Local Optimum เดิม เพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้ตำแหน่งที่ดีที่สุดจากกลุ่มอื่น

ขั้นตอนนี้จะเกิดขึ้นเฉพาะในกรณีการค้นหาแบบ Multi Group เท่านั้น โดยเมื่อกลุ่มใดกลุ่มหนึ่งติดใน Local Optimum จะเปลี่ยนค่าตำแหน่ง GBEST ของกลุ่มที่ติดให้เท่ากับค่า GBEST ของกลุ่มอื่นที่ได้จากการสุ่ม เพื่อให้ GBEST ของกลุ่มอื่นดึงอนุภาคของกลุ่มที่ติดให้หลุดจากการติด Local Optimum

3. การ Reposition หรือ Slightly Mutation

การ Reposition หรือ Slightly Mutation เป็นขั้นตอนการปรับเปลี่ยนตำแหน่งของอนุภาค และ GBEST ที่นำมาใช้ในการแก้ปัญหาการติดใน Local Optimum และยังเป็นโอกาสให้อนุภาคเคลื่อนที่ไปค้นหาค่าคำตอบที่บริเวณอื่นเพื่อไม่ให้อนุภาคกลับมาอยู่จุดที่ติด Local Optimum เดิม โดยมีข้อสังเกตว่าในการติด Local Optimum ค่าตำแหน่งในมิติต่างๆ มักจะมีค่าที่ตรงกับ Global Optimum จะมีเพียงบางมิติเท่านั้นที่มีค่าที่ไม่ตรงและมักจะไม่ห่างจากตำแหน่งที่เป็น Global Optimum วิทยานิพนธ์ฉบับนี้จึงได้ทำปรับเปลี่ยนตำแหน่งในการกลายพันธุ์ให้อยู่ในช่วงแคบๆคือเปลี่ยนจำนวนมิติจากค่าเดิมเพียง 20% - 30% ของจำนวนมิติทั้งหมด นอกจากนั้นในมิติที่เปลี่ยนจะเปลี่ยนจากตำแหน่งเดิมไม่เกิน 10% ซึ่งเป็นการช่วยลดเวลาในการค้นหาและลู่เข้าหาคำตอบ โดยช่วงที่เลือกมาทำการกระจายจะอยู่ในช่วง 0.9 - 1 ของค่าเดิม โดยขั้นตอนในการ Reposition แบ่งออกเป็น 2 ส่วน คือ

3.1 การ reposition ตำแหน่งของอนุภาค

ขั้นตอนนี้จะเกิดขึ้นหลังจากอนุภาคติดอยู่ใน Local optimum ไม่ว่าจะเป็นในกรณีการค้นหาแบบ Single Group หรือ Multi Group โดยจะทำการเลือกเปลี่ยนตำแหน่งของอนุภาคเพียงแค่ 20% ของมิติทั้งหมดของอนุภาค ซึ่งจะมีการเลือกเปลี่ยนเครื่องหมายของอนุภาค 50% ของอนุภาคที่ถูกเลือก ซึ่งแสดงตามสมการที่ (4.1) โดยใช้เครื่องหมาย \pm

$$X_{id} = \pm(X_{id} \times (0.9 + 0.1 \times rand())) \quad (4.1)$$

เอกสารนี้เป็นเอกสารที่สงวน ในกรณีที่มีการเปลี่ยนเครื่องหมายจาก + ไป - หรือ - ไป + ซึ่งจะเกิดใน 50% ของค่า มิติที่เปลี่ยนนั้นก็คือ 10% ของมิติทั้งหมด ในมิติเหล่านี้จะเป็นการเปลี่ยนอย่างมากซึ่งจะไม่เปลี่ยนอย่างแคบ(Slightly Mutation) ตามที่กำหนดโดยค่า $0.9 + 0.1 * random()$

3.2 การ reposition ตำแหน่ง GBEST

มีขั้นตอนที่ทำเช่นเดียวกับการ Reposition ของตำแหน่งอนุภาคเพียงแต่จำนวนมิติที่จะมีการเปลี่ยนแปลงเพิ่มขึ้นเป็น 30% เนื่องจากมีการ Reset ค่า PBEST และ GBEST ก่อนที่จะทำขั้นตอน 3.1 และ 3.2 ดังนั้นในการทำขั้นตอน 3.1 ค่า GBEST จะเป็นตำแหน่งเดียวกับอนุภาคที่ดีที่สุดหลังการ Reposition จากนั้นในขั้นตอนนี้จะสุ่มเปลี่ยนค่าเพิ่มเติมจากเดิมอีก 30% ของมิติทั้งหมด ดังนั้นจริงๆแล้วจำนวนมิติที่เปลี่ยนแปลงจากค่า GBEST เดิมมากที่สุดคือ 50% ของจำนวนมิติทั้งหมด (โดยมีข้อสมมุติว่าตำแหน่งอนุภาคและตำแหน่ง GBEST อยู่ใกล้เคียงกันตอนที่ติด Local Optimum โดย 50% ได้มาจาก 20% ตอน Reposition ตำแหน่งอนุภาค และอีก 30% จากการที่นำอนุภาคนี้มา Reposition เป็น GBEST) ตามสมการที่ (4.2) และทำการเลือกเปลี่ยนเครื่องหมายของอนุภาค GBEST 50% ของอนุภาคที่ถูกเลือก

$$G_{id} = \pm(G_{id} \times (0.9 + 0.1 \times rand())) \quad (4.2)$$

กระบวนการนี้สามารถเกิดขึ้นได้ทั้งในกรณีที่กลุ่มอนุภาคเป็นแบบ Single Group และ Multi Group โดยใน Multi Group จะเกิดขึ้นเฉพาะในกรณีที่ทุกกลุ่มติดอยู่ใน local optimum เดียวกันเท่านั้น

4. การลดจำนวนกลุ่มอนุภาค

กระบวนการลดจำนวนกลุ่มอนุภาคเป็นขั้นตอนที่เพิ่มความสามารถในการค้นหาแบบเฉพาะที่หรือ Local Search โดยกระบวนการค้นหาแบบ Single Group หรือ ลดจำนวนกลุ่มอนุภาคเหลือเพียง 1 กลุ่มอนุภาคจะเกิดขึ้นเมื่อกระบวนการค้นหาคำคำตอบแบบ Multi Group ทำการรีโพลิชชันเมื่อติดใน Local Optimum ครบจำนวน 4 รอบ ซึ่งการลดจำนวนกลุ่มนี้มีวัตถุประสงค์เพื่อเพิ่มความสามารถในการค้นหาแบบเฉพาะที่ให้กับอนุภาค ทำให้อนุภาคสามารถหาคำคำตอบได้ลึกขึ้น

โครงสร้างข้อมูลของการแบ่งกลุ่มของอนุภาคจะใช้โครงสร้าง Array โดยถ้าเริ่มต้นอนุภาคมี 90 อนุภาค ถ้าแบ่งออกเป็น 3 กลุ่ม อนุภาคไหนจะอยู่ที่กลุ่มใดจะใช้การ Modulo Index ของอนุภาค โดยกลุ่มของอนุภาคจะเท่ากับ Index ของอนุภาค Modulo 3 ดังนั้น Swarm[0], Swarm[3] และถ้าไม่ว่ากรณี Swarm[6] จะอยู่ในกลุ่มเดียวกันเนื่องจาก Index Modulo แล้วเท่ากับ 0 การใช้โครงสร้างนี้ทำให้เปลี่ยนจำนวนกลุ่มได้ง่ายโดยถ้าต้องการเปลี่ยนจาก 3 กลุ่มเป็น 2 กลุ่มก็แค่เปลี่ยนจำนวนที่ไป

Modulo ของ Index จาก 3 เป็น 2 นอกจากนี้วิธีนี้มีข้อดีคือตำแหน่งอนุภาคที่อยู่ในกลุ่มเดิมสามารถส่งต่อบางส่วนเข้าไปในกลุ่มใหม่หลังจากการ Reposition ซึ่งเพิ่มการสื่อสารระหว่างกลุ่มนอกจากการใช้ผ่านทาง GBEST อย่างเดียว

5. การเพิ่มจำนวนกลุ่มอนุภาค

กระบวนการเพิ่มจำนวนกลุ่มอนุภาคเป็นขั้นตอนที่เพิ่มความสามารถในการค้นหาในแนวกว้าง หรือ Global Search จะเกิดขึ้นเมื่อกลุ่มอนุภาค 1 กลุ่ม ทำการรีโพอิชชันเมื่อติดในจุด Local Optimum ทั้งหมด 6 ครั้ง จากนั้นทำการกระจายอนุภาคออกเป็นกลุ่มย่อยๆ ในงานวิจัยนี้คือจำนวน 3 กลุ่ม โดยการเพิ่มจำนวนกลุ่มอนุภาคนี้มีวัตถุประสงค์เพื่อเพิ่มพื้นที่ในการค้นหาทำให้อนุภาคสามารถค้นหาคำตอบได้กว้างขึ้น

กระบวนการทำงานของวิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดใน Local Optimum ของ PSO หรือ MSPSO แสดงดัง รูปที่ 4.5 และ 4.6 และแผนผังการทำงานแสดงดังรูปที่ 4.7

Initial swarm (90,1,X[],V[],GBEST,PBEST[])/number of particles, number of swarm, particles positions	(1)
While(not finish)	(2)
Iteration++	(3)
If number of swarm = 1	(4)
for all particle p in a swarm	(5)
Update particles p position using normal PSO from EQ (2.2,2.3)	(6)
Evaluate particle p positions and update its PBEST and GBEST	(7)
If GBEST unchanged for Max_ iterations(100):	(8)
reset GBEST	(9)
for all particle p in a swarm	(10)
reset p's PBEST value to maximum	(11)
randomly mutate p positions by EQ[4.1] //20% of p's dimension	(12)
Evaluate particle p positions and update its PBEST and GBEST	(13)
Set GBEST to randomly mutate GBEST positions by EQ[4.2] //30%	(14)
Number of single group Reposition ++	(15)
If Number of single group Reposition = 6	(16)
Set number of swarm to 3 //divide a swarm into 3 sub-swarm	(17)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.5 รหัสจำลองแสดงการทำงานของ MSPSO ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรหัสจำลองการแสดงผลการทำงานของ MSPSO รูปที่ 4.5 ซึ่งเป็นการทำงานของอัลกอริทึมใน ส่วนการค้นหาแบบ Single Group โดยในช่วงเริ่มต้นการทำงานจะทำการค้นหาแบบ PSO ต้นแบบ จนเมื่อมีอนุภาคติดใน Local Optimum จะเข้าสู่กระบวนการ Reposition แบบ Single Group ของ MSPSO ซึ่งจะเห็นได้จากบรรทัดที่ 8-15 คือทำการ Reset ค่า GBEST และ PBEST จากนั้นทำการ Reposition ตำแหน่งของอนุภาค p ตามสมการที่ 4.1 และทำการ Reposition ตำแหน่งของ GBEST ตามสมการที่ 4.2 และวนกลับไปทำการค้นหาแบบ PSO ต้นแบบอีกครั้ง โดยกระบวนการ Reposition แบบ Single Group นี้จะทำทั้งหมด 6 ครั้ง จากนั้นอนุภาคจะทำการแบ่งกลุ่มอนุภาค ออกเป็น 3 กลุ่ม ดังบรรทัดที่ 16-17 และเริ่มต้นการค้นหาแบบ Multi Group ดังจะแสดงในรูปที่ 4.6

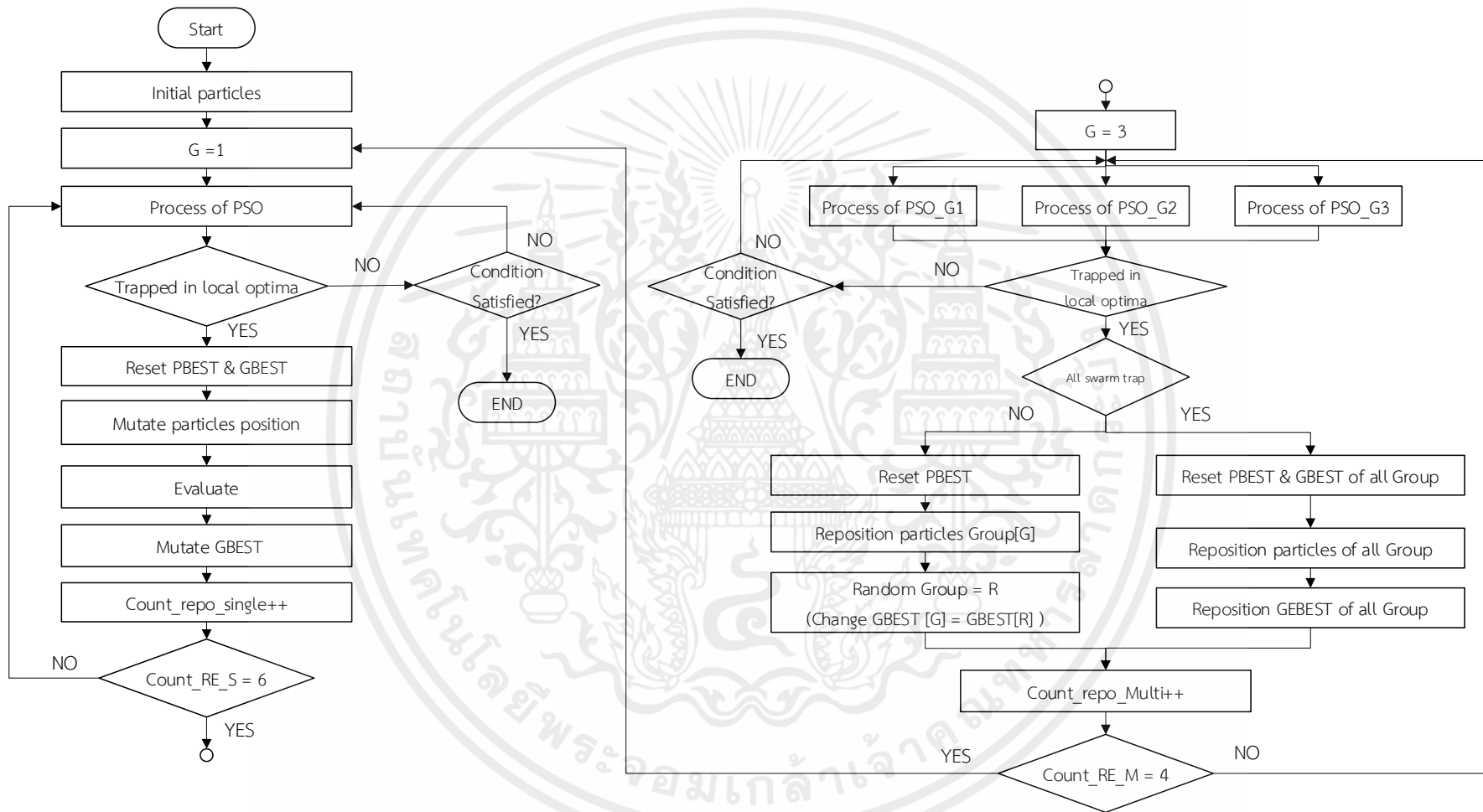
Else (number of swarm \neq 1)	(18)
for all sub swarm s in a swarm	(19)
for all particle p in a sub swarm s	(20)
Update particles p position using normal PSO from EQ (2.2,2.3)	(21)
Evaluate particles p and update p's PBEST and sub swarm s GBEST	(22)
If all sub-swarm trap in the same local optima:	(23)
for all sub swarm s in a swarm	(24)
reset sub swarm s GBEST	(25)
for all particle p in a sub swarm s	(26)
reset particle p PBEST	(27)
randomly mutate particle p positions by EQ[4.1]	(28)
evaluate particle p positions, update its PBEST, sub swarm s GBEST	(29)
randomly mutate sub swarm s GBEST using EQ[4.2]	(30)
Else (not all sub-swarm trap in the same local optima):	(31)
for all particle p in a trap sub swarm s	(32)
reset particle p PBEST	(33)
randomly mutate particle p positions by EQ[4.1]	(34)
evaluate particle p positions, update its PBEST	(35)
set GBEST of a trap swarm s to GBEST of other randomly sub swarm with -	(36)
different GBEST	
Number of multi group Reposition ++	(37)
If Number of multi group Reposition = 4	(38)
Set number of swarm to 1	(39)

รูปที่ 4.6 รหัสจำลองแสดงผลการทำงานของ MSPSO (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ โดยผู้จัดทำเอกสารนี้ไว้เพื่อใช้ในการศึกษาวิจัยเท่านั้น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรหัสจำลองการแสดงผลการทำงานของ MSPSO รูปที่ 4.6 ซึ่งเป็นการทำงานของอัลกอริทึมใน ส่วนของการค้นหาแบบ Multi Group โดยในช่วงเริ่มต้นการทำงาน แต่ละกลุ่มจะทำการค้นหาแบบ PSO ต้นแบบเป็นอิสระแยกออกจากกัน แต่ละกลุ่มมีค่า PBEST และ GBEST เป็นของตนเอง จนเมื่อมีการติดใน Local Optimum จะเข้าสู่กระบวนการ Reposition ซึ่งแบ่งออกเป็น 2 กรณี คือ ถ้าทุกกลุ่มติด Local Optimum ที่จุดเดียวกัน (บรรทัดที่ 23 - 30) แต่ละกลุ่มจะทำ Reset ค่า GBEST และ PBEST ก่อน จากนั้นทำการ Reposition ตำแหน่งของอนุภาค p ของแต่ละกลุ่มตามสมการที่ 4.1 และทำการ Reposition ตำแหน่งของ GBEST ของแต่ละกลุ่มตามสมการที่ 4.2 แต่ถ้ามีเพียงแค่บางกลุ่ม เท่านั้นที่ติดใน Local Optimum (บรรทัดที่ 31-37) จะทำการ Reposition เฉพาะกลุ่มที่ติดเท่านั้น โดยจะเริ่มจาก Reset ค่า PBEST ของกลุ่มที่ติด จากนั้นทำการ Reposition ตำแหน่งของอนุภาค p ตามสมการที่ 4.1 ส่วนค่า GBEST ปรับปรุงโดยเปลี่ยนค่าตำแหน่ง GBEST ของกลุ่มที่ติดให้เท่ากับค่า GBEST ของกลุ่มอื่นที่ได้จากการสุ่ม เพื่อให้ GBEST ของกลุ่มอื่นตั้งอนุภาคของกลุ่มที่ติดให้หลุดจากการติด Local Optima ซึ่งกระบวนการ Reposition แบบ Multi Group จะทำทั้งหมด 4 ครั้ง จากนั้น จะทำการรวมกลุ่มอนุภาคเป็น 1 กลุ่ม (บรรทัดที่ 38 - 39) และวนกลับไปค้นหาแบบ Single Group (รูปที่ 4.5) ต่อไป โดยทั้งนี้ขั้นตอนการค้นหาของ MSPSO นี้ จะทำงานสลับกันไปมา ระหว่างแบบ Single Group กับ Multi Group จนกว่าจะเจอค่าคำตอบที่ดีที่สุดหรือครบรอบการค้นหาที่กำหนดไว้ นั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แผนผังการทำงานของ MSPSO

บทที่ 5

การทดลอง

ในบทนี้ได้กล่าวถึงการทดลองและผลการทดลองของงานวิจัยนี้และงานวิจัยก่อนหน้าที่นำมาเปรียบเทียบเพื่อวัดประสิทธิภาพในการหาผลลัพธ์ค่าที่เหมาะสมที่สุดของฟังก์ชันมาตรฐานทั้งหมด 27 ฟังก์ชัน โดยการทดลองได้มีการใช้วิธีของงานวิจัยต่างๆดังต่อไปนี้

1. วิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization, PSO[1,2])
2. วิธีการค้นหาอย่างรวดเร็วแบบหลายกลุ่มอนุภาคโดยการกลายพันธุ์และการผสมพันธุ์ (Fast Multi-swarm Particle Swarm Optimization, FMPSO[16])
3. วิธีการค้นหาขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบหลากหลายกลุ่มโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ร่วมกับการเปลี่ยนตำแหน่งของอนุภาคที่ดีที่สุดของกลุ่ม (Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm, MPSOSM[7])
4. ขั้นตอนวิธีการปรับปรุงหาค่าความเหมาะสมที่สุดแบบกลุ่มอนุภาคโดยใช้การกระจายตำแหน่งของอนุภาคเมื่อติดในจุดสุดสัมพัทธ์ (Reposition by Slightly Mutation Particle with Particle Swarm Optimization, RPSO)
5. วิธีการเริ่มต้นการทำงานใหม่ของการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาคด้วยการกลายพันธุ์ของตำแหน่งที่ดีที่สุดของอนุภาค (The use of Global Best position in rerun of particle swarm optimization, GBPSO[8])
6. ขั้นตอนวิธีการรวมกลุ่มอนุภาคแบบหลากหลายกลุ่มเพื่อหาค่าความเหมาะสม (Combine multi particle swarm in supporting trapping in local optima, CMPSO[9])
7. วิธีการรวมและกระจายกลุ่มอนุภาคเพื่อแก้ปัญหาการติดใน Local Optimum ของ PSO (Merge and split particle swarm in reducing trapping in local optima of PSO, MSPSO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุผลบางเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 ค่าพารามิเตอร์

ค่าพารามิเตอร์ที่นำมาใช้ในการเปรียบเทียบจะถูกกำหนดให้มีค่าเท่ากับของทดลองของงานวิจัยนี้ในทุกอัลกอริทึม ซึ่งจะถูกกำหนดไว้ดังตารางที่ 5.1 ด้านล่าง

ตารางที่ 5.1 ค่าพารามิเตอร์พื้นฐานที่ใช้ทดลองกับฟังก์ชันมาตรฐาน

พารามิเตอร์	ค่าพารามิเตอร์
จำนวนอนุภาค (Population Size) ที่ใช้ในการทดลองกรณีที่เป็น 1 กลุ่มอนุภาค	90 อนุภาค
จำนวนกลุ่มของอนุภาคที่ใช้ในการทดลองกรณีที่เป็นหลายกลุ่มอนุภาค	3 กลุ่ม
จำนวนอนุภาค (Population Size) ที่ใช้ในการทดลองกรณีที่เป็นหลายกลุ่มอนุภาค/กลุ่ม	30 อนุภาค/กลุ่ม
จำนวนรอบที่ใช้ในการประมวลผลการทดลอง	10 ครั้ง
ค่าความเฉื่อย (Inertia Weight; ω)	0.72984
ค่าคงที่การเรียนรู้ (Acceleration Evaluation; $C1=C2$)	1.496172
จำนวนมิติของแต่ละฟังก์ชัน (Dimension)	ตามตารางที่ 5.2
จำนวนรอบมากที่สุดที่ใช้ในการทดลอง (Generation)	1,000,000 รอบ
จำนวนรอบที่นับว่าติดใน Local Optima (Threshold of trapped)	100
จำนวนครั้งในการกระจายของอนุภาค(Threshold of Reposition) กรณีมีการทำงานแบบ 1 กลุ่มอนุภาค	6
จำนวนครั้งในการกระจายของอนุภาค(Threshold of Reposition) กรณีมีการทำงานแบบหลายกลุ่มอนุภาค	4

5.2 การทดลอง

งานวิจัยนี้ได้ทำการทดลองเปรียบเทียบเพื่อวัดประสิทธิภาพระหว่างอัลกอริทึม PSO, RPSO, GBPSO, FMPSO, MPSOSM และ CMPSO กับอัลกอริทึมที่นำเสนอมานี้ (MSPSO) โดยทดสอบบนฟังก์ชันมาตรฐาน(Benchmark test function) จำนวน 27 ฟังก์ชัน (ภาคผนวก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ฟังก์ชันที่ใช้ในการทดลอง

โดยในงานวิจัยนี้ทำการทดลองบนฟังก์ชันมาตรฐานทั้งหมด 27 ฟังก์ชัน (ภาคผนวก) โดยแบ่งเป็นฟังก์ชัน Multimodal จำนวน 18 ฟังก์ชัน และ Unimodal จำนวน 9 ฟังก์ชัน ที่มีจำนวนมิติตามตารางด้านล่าง

ตารางที่ 5.2 ลักษณะและจำนวนมิติของฟังก์ชัน

ชื่อฟังก์ชัน	มิติ	ลักษณะของฟังก์ชัน
ACKLEY	100	Multimodal
GRIEWANK	100	Multimodal
RASTRIGIN	100	Multimodal
ROSENBROCK	100	Multimodal
SCHWEFEL	100	Multimodal
COSINE MIXTURE	100	Multimodal
EXPONENTIAL	100	Multimodal
LEVY	100	Multimodal
MICHALEWICZ	10	Multimodal
DIXON-PRICE	10	Multimodal
STEP	100	Multimodal
SCHAFFER	2	Multimodal
HOLDER	2	Multimodal
BEALE	2	Multimodal
SHUBERT	2	Multimodal
GOLDSTEIN-PRICE	2	Multimodal
SIX-HUMP CAMEL	2	Multimodal
SPHERE	100	Unimodal
PARALLEL HYPER-ELLIPSOID	100	Unimodal
ROTATED HYPER-ELLIPSOID	100	Unimodal
CIGAR	100	Unimodal
BROWN	100	Unimodal
MULTIMOD	100	Unimodal
ZAKHAROV	100	Unimodal
TRID	10	Unimodal
EASOM	2	Unimodal
EGGHOLDER	2	Multimodal

5.4 ผลการทดลอง

ผลการทดลองตามข้อที่ 5.2 จะแบ่งตารางการแสดงผลออกเป็น 2 ส่วน ส่วนแรกเป็นการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่ใช้ 1 กลุ่มอนุภาคในการค้นหาค่าคำตอบซึ่งจะแสดงในตารางที่ 5.3 และ 5.4 ส่วนที่สองเป็นการเปรียบเทียบระหว่างอัลกอริทึมที่ใช้หลายกลุ่มอนุภาคในการค้นหาค่าคำตอบซึ่งจะแสดงในตารางที่ 5.5 และ 5.6 โดยทั้งสองส่วนทำการทดสอบบนฟังก์ชันมาตรฐานทั้ง 27 ฟังก์ชันในหัวข้อ 5.3 โดยรายละเอียดของแต่ละฟังก์ชันอธิบายในภาคผนวก ก ซึ่งใช้ค่าพารามิเตอร์ตามตารางที่ 5.1 และ 5.2

ตารางที่ 5.3 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานแบบ 1 กลุ่มอนุภาค PSO, RPSO, GBPSO และที่นำเสนอที่เป็นแบบหลายกลุ่มอนุภาค CMPSO กับMSPSO

NO	Function	PSO	RPSO	GBPSO(30)*	GBPSO(90)	CMPSO	MSPSO
		Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call
1	ACKLEY	30,115,176	5,106,860	1,650,657	3,135,697	<u>2,958,060</u>	2,930,509
2	GRIEWANK	5,290,716	471,592	290,627	460,581	391,325	<u>383,057</u>
3	RASTRIGIN	13,997,841	2,442,045	2,035,588	3,836,991	<u>3,169,668</u>	3,947,736
4	ROSENBROCK	724,874	90,048,115	8,185,827	1,171,471	1,097,672	<u>1,006,174</u>
5	SCHWEFEL	51,283,412	43,860,919	28,803,838	28,652,968	29,740,709	<u>25,644,435</u>
6	COSINE MIXTURE	47,399,480	556,134	333,556	<u>483,072</u>	496,443	489,148
7.	EXPONENTIAL	5,064,847	1,249,951	333,145	731,974	682,240	<u>693,692</u>
8.	LEVY	1,117,264	3,412,657	3,643,987	2,397,063	2,280,170	<u>1,933,746</u>
9.	MICHALEWICZ	9,490,158	<u>113,034</u>	77,111	176,389	119,337	163,624
10.	DIXON-PRICE	<u>8,034</u>	17,213	24,202	13,317	10,016	9,212
11.	STEP	81,980,437	<u>109,751</u>	63,111	123,529	121,235	120,291
12.	SCHAFFER	<u>20,380</u>	24,481	53,955	171,236	140,682	52,690
13.	HOLDER	<u>7,097</u>	14,857	19,867	17,822	18,261	13,267
14.	BEALE	<u>30,538</u>	31,915	11,803	32,291	32,317	32,353
15.	SHUBERT	6,589	5,857	2,439	6,248	6,061	<u>5,745</u>
16.	GOLDSTEIN-PRICE	11,798	11,636	4,233	12,025	11,949	<u>10,919</u>
17.	SIX-HUMP CAMEL	<u>1,961</u>	2,241	931	2,197	2,412	2,081
18	SPHERE	64,780,130	5,397,412	4,294,113	5,435,301	5,412,248	<u>5,406,769</u>
19.	Parallel	27,382,808	5,417,563	4,365,020	5,438,400	5,453,967	<u>5,398,414</u>
20.	Rotated	6,137,700	5,388,812	4,302,733	5,371,351	5,348,542	<u>5,333,830</u>

NO	Function	PSO	RPSO	GBPSO(30)*	GBPSO(90)	CMPSO	MSPSO
		Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call
21.	CIGAR	8,360,121	<u>5,407,012</u>	4,380,980	5,442,806	5,407,171	5,428,116
22.	BROWN	50,726,208	5,386,134	4,387,605	<u>5,373,773</u>	5,385,187	5,416,673
23.	MULTIMOD	139,223	15,426,328	11,342,805	14,777,380	<u>14,356,234</u>	14,522,362
24.	ZAKHAROV	88,352,774	85,329,067	29,999,950	88,409,486	<u>84,659,455</u>	84,884,260
25.	TRID	28,234,935	1,530,782	147,171	360,485	404,130	<u>299,442</u>
26.	EASOM	<u>14,095</u>	17,917	6,710	17,415	17,581	18,085
27.	EGGHOLDER	18,594	84,800,595	30,172,287	<u>83,461,070</u>	90,547,904	84,394,263

ตารางที่ 5.4 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานแบบ 1 กลุ่มอนุภาค PSO, RPSO, GBPSO และที่นำเสนอที่เป็นแบบหลายกลุ่มอนุภาค CMPSO กับ MSPSO (ต่อ)

NO	Function	PSO	RPSO	GBPSO(30)*	GBPSO(90)	CMPSO	MSPSO
		BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS
1	ACKLEY	4.56869	0	0	0	0	0
2	GRIEWANK	0.150842	0	0	0	0	0
3	RASTRIGIN	395.992	0	0	0	0	0
4	ROSENBROCK	1.53E-27	2.9336E-29	0	0	0	0
5	SCHWEFEL	6612.63	0	262.9399	0	0	0
6	COSINE MIXTURE	4.28574	0	0	0	0	0
7.	EXPONENTIAL	5.55E-16	0	0	0	0	0
8.	LEVY	96.7221	0	0	0	0	0
9.	MICHALEWICZ	0.442787	0	0	0	0	0
10.	DIXON-PRICE	0	0	0	0	0	0
11.	STEP	37	0	0	0	0	0
12.	SCHAFFER	0	0	0	0	0	0
13.	HOLDER	0	0	0	0	0	0
14.	BEALE	0	0	0	0	0	0
15.	SHUBERT	0	0	0	0	0	0
16.	GOLDSTEIN-PRICE	0	0	0	0	0	0
17.	SIX-HUMP CAMEL	0	0	0	0	0	0
18.	SPHERE	0	0	0	0	0	0
19.	Parallel	3.96241e-321	0	0	0	0	0

NO	Function	PSO	RPSO	GBPSO(30)*	GBPSO(90)	CMPSO	MSPSO
		BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS
20.	Rotated	0	0	0	0	0	0
21.	CIGAR	1.4822e-318	0	0	0	0	0
22.	BROWN	9.88131e-324	0	0	0	0	0
23.	MULTIMOD	449.13	0	0	0	0	0
24.	ZAKHAROV	5.92879e-323	2.638E-290	3.41E-38	7.3418e-321	0	0
25.	TRID	2.73E-12	0	0	0	0	0
26.	EASOM	0	0	0	0	0	0
27.	EGGHOLDER	65.0611	0.00172513	0.025918	0.0030044	0.004243	0.003645

จากตารางที่ 5.3 จะแสดงจำนวนครั้งที่ใช้ในการหาค่าต่ำสุดของฟังก์ชัน(GBEST)ที่แต่ละวิธีใช้ ณ จุดที่พบ GBEST ค่า GBEST นี้อาจไม่ใช่ค่าที่เป็นคำตอบของฟังก์ชันแต่เป็นค่าที่ดีที่สุดที่แต่ละวิธีหาได้ในจำนวนรอบมากที่สุดที่ได้กำหนดไว้คือ 1 ล้านรอบ เนื่องจากมีอนุภาค 90 อนุภาค จำนวนครั้งมากที่สุดที่ใช้ในการเรียกใช้ฟังก์ชันประเมินค่าจะอยู่ประมาณ 90 ล้านครั้ง จำนวนครั้งที่เกินจะมาจากการทำการ Reposition ของตำแหน่งอนุภาคในวิธีที่มีการ Reposition ถ้าเห็นค่าในตารางนี้เกิน 90 ล้านครั้งแสดงว่าวิธีนั้นๆทำงานไปจนครบจำนวนรอบมากที่สุดและคงไม่เจอตำแหน่งที่ดีที่สุด คือมีการค้นพบค่า GBEST ใหม่อยู่เรื่อยๆ(ไม่ติดใน Local Optimum) แต่ถ้ามีค่าน้อยกว่า 90 ล้านแสดงว่าวิธีนั้นๆหาค่าที่ดีที่สุด(GBEST)ได้และหลังจากนั้นไม่มีการเปลี่ยนค่า GBEST จนทำงานไปครบรอบที่กำหนด การที่จะดูว่าวิธีไหนเจอค่าต่ำสุดจึงต้องใช้ผลในตารางที่ 5.4 ประกอบด้วย เนื่องจากค่าที่ดีที่สุด ในตาราง 5.3 อาจเป็นวิธีที่ติด Local Optimum ที่เร็วสุดและเมื่อทำงานไปจนครบจำนวนรอบมากที่สุดที่กำหนดแต่ไม่เจอคำตอบที่ดีที่สุด(Global Optimal)ของแต่ละฟังก์ชัน

จากทั้งสองตารางจะเห็นว่า PSO สามารถหาค่าคำตอบที่ดีที่สุดได้ทั้งหมด 10 ฟังก์ชัน เนื่องจากวิธีของ PSO นั้นอนุภาคจะเคลื่อนที่หาค่าคำตอบไปจนติดอยู่ใน Local Optimum และยังคงทำการค้นหาต่อไปจนกระทั่งครบรอบที่กำหนดเอาไว้ แต่เวลาที่ใช้โดยนับจากจำนวนครั้งที่ใช้ในการประเมินค่าตำแหน่งของอนุภาคนั้นมากกว่าวิธีอื่น ส่วน RPSO จะทำการกระจายตำแหน่งของอนุภาคเมื่ออนุภาคติดอยู่ในจุด Local Optimum และมีการ Reset ค่า GBEST ทำให้ GBEST ที่ได้หลังจากการ Reposition คือตำแหน่งของอนุภาคที่ดีที่สุดหลังการกลายพันธุ์ซึ่งก็ยังไม่ไกลพอจาก GBEST เดิมทำให้กลุ่มอนุภาคกลับไปติดใน Local Optimum เดิม แต่ก็ยังดีกว่า PSO ที่ไม่มีการหลีกเลี่ยงการติดใน Local Optimum เลยจึงทำให้ RPSO สามารถหาค่าคำตอบที่ดีกว่า PSO ได้มากและ

เจอคำตอบที่ดีที่สุดทั้งหมด 24 ฟังก์ชัน ในขณะที่ GBPSO สามารถหาค่าคำตอบที่ดีที่สุดได้ทั้งหมด 25 ฟังก์ชัน เนื่องจาก GBPSO ทำการหลีกเลี่ยงการติดใน Local Optimum ด้วยการกระจายตำแหน่งของอนุภาคและ GBEST เมื่ออนุภาคมีการติดใน Local Optimum ทำให้ GBPSO มีโอกาสเจอคำตอบได้มากกว่าโดยจะเพิ่มเวลาในการค้นหาหลังการทำ Reposition ในขณะที่ PSO จะติด Local Optimum เร็วกว่าและได้ค่าที่แย่กว่า แต่เนื่องจากยังไม่พบคำตอบและยังไม่ถึงจำนวนรอบมากที่สุดจึงให้ PSO รันต่อไปจนครบรอบที่กำหนดซึ่งทำให้ใช้เวลามากกว่าวิธีอื่น สำหรับ RPSO และ GBPSO มีผลการพบคำตอบที่ดีที่สุดใกล้เคียงกันแต่ GBPSO มักจะพบได้เร็วกว่า แต่ถ้าเราลดจำนวนอนุภาคของ GBPSO เหลือ 30 อนุภาคจะทำให้การค้นหาเร็วขึ้น แต่ในฟังก์ชันที่หาค่าคำตอบได้ยากจะหาค่าคำตอบไม่เจอขณะที่ในฟังก์ชันอื่นๆจะพบคำตอบเร็วขึ้นมาก โดยฟังก์ชันที่ไม่พบจะเพิ่มขึ้นอีก 1 ฟังก์ชันก็คือ ฟังก์ชัน Schwefel ส่วนในฟังก์ชัน Rosenbrock นั้นยังคงเจอคำตอบแต่ใช้เวลาในการค้นหานานขึ้นกว่าแบบที่เป็น 90 อนุภาคดังแสดงในตารางด้านบน

CMPSO เป็นอัลกอริทึมที่เริ่มต้นการค้นหาแบบหลายกลุ่มอนุภาคเมื่อกลุ่มหนึ่งกลุ่มใดติด Local Optimum จะทำการ Reposition และใช้ตำแหน่ง GBEST ของกลุ่มอื่นเป็นตัวชักนำอนุภาค และเมื่อทุกกลุ่มติด Local Optimum ที่จุดเดียวกันจะทำการยุบรวมเป็น 1 กลุ่มอนุภาค และทำการเคลื่อนที่แบบ PSO ไปจนกว่าจะติด Local Optimum จึงทำการ Reposition นอกจากนี้ยังใช้เทคนิคการกลายพันธุ์ตำแหน่งของอนุภาคและ GBEST เข้ามาช่วยในกรณีที่มีการติดใน Local Optimum โดยจะทำแบบ 1 กลุ่มไปเรื่อยๆจนกว่าจะครบจำนวนรอบมากที่สุดตามที่กำหนด ซึ่ง CMPSO สามารถหาค่าคำตอบได้ 26 ฟังก์ชันเพิ่มจาก GBPSO 1 ฟังก์ชัน แต่มักจะเจอคำตอบที่ดีที่สุดเร็วกว่า GBPSO และ RPSO โดยวิธีที่เจอคำตอบเร็วที่สุดจะแสดงโดยการขีดเส้นใต้

MSPSO สามารถหาค่าคำตอบที่ดีที่สุด 26 ฟังก์ชันซึ่งเท่ากับ CMPSO แต่ MSPSO เจอคำตอบที่ดีที่สุดเร็วกว่าวิธีอื่นๆ เนื่องจาก MSPSO ทำการเริ่มต้นการค้นหาแบบ 1 กลุ่มอนุภาคทำให้ในช่วงแรกของการค้นหา MSPSO ลู่เข้าหาค่าคำตอบได้ดีกว่าและยังคงใช้ความสามารถในการค้นหาแบบ Local Search ควบคู่ไปกับการค้นหาแบบ Global Search เนื่องจากการมีการรวมและกระจายกลุ่มอนุภาค นอกจากนี้ยังใช้เทคนิคการกลายพันธุ์ตำแหน่งของอนุภาค, ตำแหน่งของ GBEST และการใช้ตำแหน่งของ GBEST จากกลุ่มอื่น เพื่อแก้ปัญหาการติดใน Local Optimum โดยวิธีที่

แนะนำเสนอทั้งหมดไม่มีวิธีใดเลยที่หาค่าที่ดีที่สุดของฟังก์ชัน EGGHOLDER ได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ซึ่งมีเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.5 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานหลายกลุ่มอนุภาค FMP SO, MPSOSM, และ CMP SO กับอัลกอริทึมที่นำเสนองาน (MSP SO)

NO	Function	FMP SO	MPSOSM	CMP SO	MSP SO
		Evaluation Call	Evaluation Call	Evaluation Call	Evaluation Call
1	ACKLEY	270,000,270	9,976,302	2,958,060	<u>2,930,509</u>
2	GRIEWANK	2,481,354	770,810	391,325	<u>383,057</u>
3	RASTRIGIN	270,000,270	4,345,610	<u>3,169,668</u>	3,947,736
4	ROSENBROCK	10,501,839	90,091,792	<u>1,097,672</u>	1,006,174
5	SCHWEFEL	270,000,270	61,064,589	29,740,709	<u>25,644,435</u>
6	COSINE MIXTURE	28,729,593	1,201,142	496,443	<u>489,148</u>
7.	EXPONENTIAL	270,000,270	1,296,088	<u>682,240</u>	693,692
8.	LEVY	270,000,270	14,123,690	2,280,170	<u>1,933,746</u>
9.	MICHALEWICZ	<u>4,698</u>	160,782	119,337	163,624
10.	DIXON-PRICE	39,366	11,273	10,016	<u>9,212</u>
11.	STEP	853,497	168,154	121,235	<u>120,291</u>
12.	SCHAFFER	55,890	37,754	140,682	<u>52,690</u>
13.	HOLDER	<u>270</u>	24,890	18,261	13,267
14.	BEALE	<u>28,134</u>	34,733	32,317	32,353
15.	SHUBERT	835,137	5,752	6,061	<u>5,745</u>
16.	GOLDSTEIN-PRICE	<u>7,587</u>	12,091	11,949	10,919
17.	SIX-HUMP CAMEL	2,430	2,375	2,412	<u>2,081</u>
18	SPHERE	29,501,145	12,751,789	5,412,248	<u>5,406,769</u>
19.	Parallel	29,886,219	12,822,032	5,453,967	<u>5,398,414</u>
20.	Rotated	29,239,218	12,546,220	5,348,542	<u>5,333,830</u>
21.	CIGAR	29,677,239	12,788,264	<u>5,407,171</u>	5,428,116
22.	BROWN	29,484,621	12,845,110	<u>5,385,187</u>	5,416,673
23.	MULTIMOD	270,000,270	33,288,076	<u>14,356,234</u>	14,522,362
24.	ZAKHAROV	270,000,270	90,000,307	<u>84,659,455</u>	84,884,260
25.	TRID	2,447,145	2,722,421	404,130	<u>299,442</u>
26.	EASOM	270,000,270	18,981	<u>17,581</u>	18,085
27.	EGGHOLDER	27,006,642	<u>81,266,763</u>	90,547,904	84,394,263

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.6 ตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่มีการทำงานหลายกลุ่มอนุภาค FMP SO, MPSOSM, และ CMPSO กับอัลกอริทึมที่นำเสนองาน (MSPSO) (ต่อ)

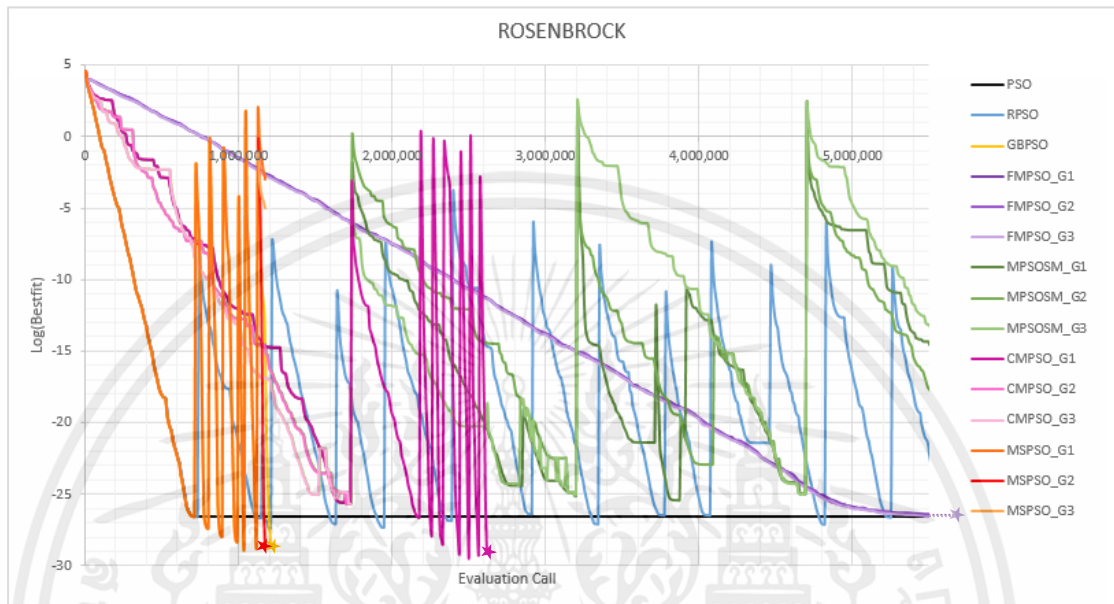
NO	Function	FMP SO	MPSOSM	CMPSO	MSPSO
		BEST FITNESS	BEST FITNESS	BEST FITNESS	BEST FITNESS
1	ACKLEY	1.42E-14	0	0	0
2	GRIEWANK	0	0	0	0
3	RASTRIGIN	744.0536	0	0	0
4	ROSENBROCK	0	3.47345E-28	0	0
5	SCHWEFEL	8402.694	12	0	0
6	COSINE MIXTURE	0.014792	0	0	0
7.	EXPONENTIAL	1.44E-16	0	0	0
8.	LEVY	0.946714	0	0	0
9.	MICHALEWICZ	0	0	0	0
10.	DIXON-PRICE	0	0	0	0
11.	STEP	0	0	0	0
12.	SCHAFFER	0	0	0	0
13.	HOLDER	0	0	0	0
14.	BEALE	0	0	0	0
15.	SHUBERT	0	0	0	0
16.	GOLDSTEIN-PRICE	0	0	0	0
17.	SIX-HUMP CAMEL	0	0	0	0
18	SPHERE	0	0	0	0
19.	Parallel	0	0	0	0
20.	Rotated	0	0	0	0
21.	CIGAR	0	0	0	0
22.	BROWN	0	0	0	0
23.	MULTIMOD	157.7624	0	0	0
24.	ZAKHAROV	9.355664	2.8602E-170	0	0
25.	TRID	0	0	0	0
26.	EASOM	1.69E-06	0	0	0
27.	EGGHOLDER	6.50611	0.0005422	0.004243158	0.003645

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น ตารางที่ 5.5 และ 5.6 เป็นตารางแสดงผลการเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่

มีการทำงานหลายกลุ่มอนุภาค FMP SO, MPSOSM, และ CMPSO กับอัลกอริทึมที่นำเสนองาน(MSPSO)

จากทั้งสองตารางจะเห็นว่า FMPSO ค้นหาค่าคำตอบที่ดีที่สุดได้ 17 ฟังก์ชัน MPSOSM สามารถหาค่าคำตอบที่ดีที่สุดได้ 24 ฟังก์ชัน ขณะที่ CMPSO และ MSPSO สามารถหาค่าคำตอบที่ดีที่สุดได้ 26 ฟังก์ชันเท่ากัน แต่ MSPSO เจอค่าคำตอบที่ดีที่สุดเร็วกว่าวิธีอื่นๆ



รูปที่ 5.1 รูปแสดงการค้นหาค่าตอบบนฟังก์ชัน ROSENBROCK ของอัลกอริทึม PSO, RPSO, GBPSO, FMPSO, MPSOSM, CMPSO กับ MSPSO

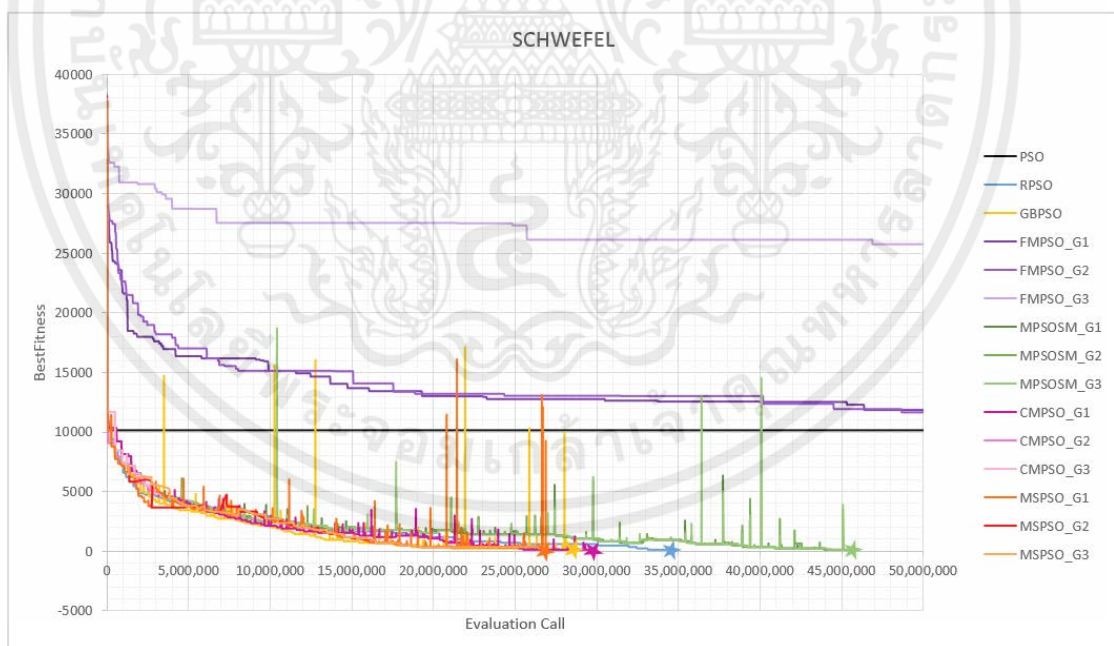
จากรูปที่ 5.1 เป็นกราฟแสดงการทำงานของฟังก์ชัน ROSENBROCK เป็นกราฟที่แสดงค่าการลู่เข้าหาค่าที่ดีที่สุดของ GBEST ของแต่ละกลุ่ม โดยที่แกน X คือ จำนวน Evaluation Call และแกน Y คือ ค่า Best Fitness ในรูปแบบของ Logarithm จากกราฟจะเห็นได้ว่าในตอนเริ่มต้น PSO หรือเส้นกราฟสีดำจะทำการลู่เข้าหาค่าคำตอบอย่างต่อเนื่องโดยในตอนต้นจะโดนทับโดยเส้นสีส้มเนื่องจากมีการทำงานช่วงต้นเหมือนกัน 4 วิธีคือ PSO, RPSO, GBPSO และ MSPSO แต่เมื่ออนุภาคทำการค้นหาไปได้ระยะเวลาหนึ่งจะเห็นว่า PSO นั้นมีค่าคงที่แม้ว่าค่า Evaluation Call จะเพิ่มขึ้นก็ตาม นั่นแสดงว่า PSO ติดอยู่ใน Local Optimum แล้วและอนุภาคยังคงทำการค้นหาค่าคำตอบอยู่ในบริเวณที่ติด Local Optimum นั้นเองโดยไม่ทำให้ค่า GBEST ดีขึ้น ส่วน RPSO หรือกราฟเส้นสีฟ้าในช่วงตอนเริ่มต้นการค้นหาอนุภาคจะลู่เข้าหาค่าคำตอบเช่นเดียวกับ PSO เนื่องจาก RPSO เป็นการค้นหาแบบ 1 กลุ่มอนุภาค จนเมื่อมีการติดใน Local Optimum RPSO จะทำการกระจายอนุภาคเพื่อทำการหลีกเลี่ยงการติดใน Local Optimum จะเห็นว่าอนุภาคมีเคลื่อนที่เข้าหา GBEST ที่ดีกว่า PSO แต่ก็ยังไม่สามารถค้นหาค่าคำตอบในฟังก์ชันนี้ได้โดยจะมีการ Reposition ไปเรื่อยๆ ส่วน GBPSO

หรือกราฟสี่เหลี่ยมเป็นการค้นหาแบบ 1 กลุ่มอนุภาคเช่นเดียวกันทำให้การลู่เข้าหาค่าตอบในช่วงแรก มีลักษณะเดียวกันกับ PSO แต่เมื่ออนุภาคมีการติดใน Local Optimum แล้ว GBPSO แก้ปัญหาโดยทำการกลายพันธุ์ตำแหน่งของอนุภาคเช่นเดียวกับ RPSO แต่เพิ่มขั้นตอนการกลายพันธุ์ตำแหน่งของ GBEST เข้าไปด้วยเพื่อป้องกันไม่ให้อนุภาคลู่เข้าหาค่า GBEST ที่ตำแหน่งเดิม ซึ่งเป็นสาเหตุให้ GBPSO สามารถเจอค่าคำตอบที่ดีที่สุดและใช้เวลาในการค้นหาได้เร็วกว่าอย่างเห็นได้ชัด แต่เนื่องจากมีการทำงานเริ่มต้นเหมือนกับวิธี MSPSO (สี่สี่) กราฟช่วงแรกจึงทับกับเส้นสี่สี่

ขณะที่ MSPSO หรืออัลกอริทึมที่นำเสนอนี้ ได้กำหนดให้เริ่มต้นการค้นหาแบบ 1 กลุ่มอนุภาค เช่นเดียวกับ PSO, RPSO และ GBPSO จึงทำให้ในช่วงเริ่มต้นการค้นหาคำตอบมีลักษณะเดียวกัน จนเมื่ออนุภาคมีการติดใน Local Optimum อนุภาคจะทำการ Reposition แบบวิธี GBPSO จำนวน 6 ครั้ง ดังนั้นจะมีการ Reposition 6 ครั้ง จึงจะเริ่มมีการทำงานที่แตกต่างกับวิธี GBPSO โดยแสดงดังกราฟสี่สี่ เมื่อทำการ Reposition ครบ 6 ครั้งแล้วแต่ยังไม่เจอคำตอบ MSPSO จึงทำการแบ่งกลุ่มอนุภาคออกเป็น 3 กลุ่ม โดยแสดงในกราฟเป็น MSPSO_G1, MSPSO_G2 และ MSPSO_G3 โดยตอนเริ่มต้นที่เป็นแบบกลุ่มเดียวก็จะใช้ MSPSO_G1 สี่สี่เช่นเดียวกัน จะเห็นว่าวิธีนี้กราฟเริ่มจากสี่สี่เส้นเดียวเมื่อมีการ Reposition ครบ 6 ครั้งจะแตกออกเป็น 3 สี และในที่สุดสี่แดง (MSPSO_G2) เจอคำตอบทำให้เส้นสี่สี่ทั้งสองที่เหลือจึงหยุดการทำงานโดยยังไม่ได้เคลื่อนที่เข้ามาใกล้คำตอบ วิธีนี้เพิ่มโอกาสในการค้นหาคำตอบให้กว้างขึ้น จะเห็นได้ว่าเมื่อมีการเพิ่มพื้นที่ในการค้นหาคำตอบแล้วอนุภาคเจอคำตอบหลังจากการแบ่งกลุ่มอนุภาคเพียงไม่กี่รอบการค้นหา ซึ่งเร็วกว่าการค้นหาแบบ GBPSO ที่เป็นการค้นหาแบบ 1 กลุ่มอนุภาค และมีจำนวนการ Reposition มากกว่า

MPSOSM, CMPSO และ FMPSO เป็นอัลกอริทึมที่เริ่มต้นการค้นหาคำตอบที่ดีที่สุดแบบหลายกลุ่มอนุภาค จากรูปที่ 5.1 จะเห็นว่า MPSOSM หรือกราฟสี่เหลี่ยมเริ่มต้นการค้นหาด้วย 3 กลุ่มอนุภาคแต่ละกลุ่มจะทำการค้นหาอิสระจากกันและมีค่า GBEST ของตัวเอง จากรูปจะเห็นได้ชัดว่าในช่วงแรกของการค้นหาคำตอบนี้ MPSOSM ลู่เข้าหาค่าคำตอบได้ช้ากว่าการค้นหาแบบ 1 กลุ่มอนุภาค MPSOSM จะทำการค้นหาคำตอบไปเรื่อยๆจนเมื่อมีการติด Local Optimum จะทำการตรวจสอบว่าถ้ามีเพียงแค่บางกลุ่มเท่านั้นที่ติด กลุ่มที่ติด Local Optimum จะทำการ Reposition ด้วยการเปลี่ยนตำแหน่งของอนุภาคและใช้ค่า GBEST จากการสุ่มจากกลุ่มข้างเคียงที่ไม่ติดใน Local Optimum แต่ถ้าทุกกลุ่มติด Local Optimum ที่จุดเดียวกัน MPSOSM จะทำการ Reposition ด้วยการเปลี่ยนตำแหน่งของอนุภาคและ GBEST ของแต่ละกลุ่มด้วยซึ่งคล้ายกับวิธี CMPSO เพียงแต่วิธี CMPSO รวมกลุ่มเป็นกลุ่มเดียว ดังนั้นเส้นกราฟทั้งสองวิธีจะทับกัน คือ เส้นกลุ่มสี่เหลี่ยมโดนทับด้วย

เส้นกลุ่มสีชมพูจนเมื่อทุกกลุ่มติดอยู่ใน Local Optimum เดียวกัน สองวิธีจะมีการค้นหาที่ต่างกัน (ประมาณ Evaluation Call ที่ 1.7 ล้าน) ซึ่งจะเห็นว่า MPSOSM ไม่สามารถค้นหาคำตอบได้ใน ฟังก์ชัน Rosenbrock ขณะที่ CMPSO หรือกราฟสีชมพูซึ่งปรับปรุงมาจาก MPSOSM ด้วยการลด จำนวนกลุ่มลงเป็นกลุ่มเดียว เมื่อทุกกลุ่มอนุภาคติดใน Local Optimum ที่จุดเดียวกัน โดย CMPSO ก็จะกลายเป็นวิธีเดียวกับ GBPSO และจะทำการค้นหาไปเรื่อยๆ หากติดใน Local Optimum อีก CMPSO จะทำการ Reposition แบบ GBPSO จากกราฟจะเห็นว่าเมื่อมีการรวมกลุ่มอนุภาคแล้ว อนุภาคสามารถค้นหาคำตอบได้และใช้เวลาในการค้นหาน้อยกว่า MPSOSM เป็นอย่างมากนั่นเอง ขณะที่ FMPSO นั้นแม้ว่าจะเป็นการค้นหาแบบหลายกลุ่มอนุภาคเช่นเดียวกันแต่ในระหว่างการค้นหา ของ FMPSO นั้นได้เพิ่มขั้นตอนการกลายพันธุ์แบบคอซีเข้ามาด้วย ซึ่งทำให้อนุภาคกระจายการค้นหา ในแนวกว้างเพื่อเพิ่มพื้นที่ในการค้นหาและป้องกันการติดใน Local Optimum แต่ก็เป็นที่เหตุให้การลู่ เข้าหาคำตอบของ FMPSO นี้ช้ากว่าแบบอื่น ๆ นั้นเอง แม้ว่า FMPSO จะลู่เข้าหาคำตอบช้ากว่าวิธีอื่น แต่สามารถหาคำตอบในฟังก์ชันนี้ได้ เนื่องจากไม่มีการ Reposition แต่ละกลุ่มของ FMPSO กลุ่มสี ม่วงจึงมีตำแหน่งที่ใกล้เคียงกันมากและแทบจะเห็นเป็นเส้นเดียว



รูปที่ 5.2 รูปแสดงการค้นหาหาคำตอบบนฟังก์ชัน SCHWEFEL ของอัลกอริทึม PSO, RPSO, GBPSO,

FMPSO, MPSOSM, CMPSO กับ MSPSO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.2 เป็นกราฟแสดงการทำงานของฟังก์ชัน SCHWEFEL เป็นกราฟที่แสดงค่าการลู่เข้าหาค่าที่ดีที่สุดของ GBEST โดยที่แกน X คือ จำนวน Evaluation Call และแกน Y คือ ค่า Best Fitness จากกราฟจะเห็นว่า PSO เมื่อค้นหาคำตอบได้ระยะเวลาหนึ่งกราฟจะคงที่หรือติดใน Local Optima และไม่สามารถหาคำตอบที่ดีขึ้นแม้ว่าจะใช้ระยะเวลาในการค้นหาไปอีกก็รอบก็ตาม ในขณะที่วิธีอื่นๆ RPSO, GBPSO, CMPSO และ MPSO มีการลู่เข้าหาคำตอบเนื่องจากการเพิ่มขึ้นตอนในการกระจายอนุภาคเข้าไปแก้ปัญหา จะเห็นว่า MPSO เจอคำตอบได้เร็วกว่าวิธีอื่น ตามด้วย CMPSO, GBPSO, RPSO และ MPSOSM ตามลำดับ ส่วน FMPSO มีการลู่เข้าหาคำตอบช้ากว่าวิธีอื่นและไม่สามารถหาค่าของคำตอบเจอเช่นเดียวกับ PSO



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการทดลองและข้อเสนอแนะ

6.1 สรุปผลการทดลอง

งานวิจัยนี้ได้นำวิธีการหาค่าที่เหมาะสมที่สุดแบบกลุ่มอนุภาคมาพัฒนาและปรับปรุงเพื่อแก้ปัญหาการติดใน Local Optimum ด้วยการกระจายกลุ่มอนุภาคโดยการกลายพันธุ์ตำแหน่งของอนุภาค, GBEST และการใช้ค่า GBEST จากกลุ่มอื่น ร่วมกับการรวมและกระจายกลุ่มอนุภาคเพื่อใช้ความสามารถในการค้นหาทั้งแบบ Global Search และ Local Search ให้เหมาะสมและมีประสิทธิภาพ

งานวิจัยนี้ได้ทำการทดลองเพื่อทดสอบประสิทธิภาพในการค้นหาผ่านฟังก์ชันมาตรฐานที่หลากหลาย ทั้งแบบ Many Local Minima และ Single Local Minima รวมทั้งหมด 27 ฟังก์ชันเทียบกับอัลกอริทึมต้นแบบ(PSO)และอัลกอริทึมอื่นๆที่มีลักษณะใกล้เคียงกัน (RPSO, GBPSO, FMPSO, MPSOSM และ CMPSO) จากการทดลองในบทที่ 5 หัวข้อที่ 5.4 ตามตารางที่ 5.3 – 5.6 จะเห็นว่า MSPSO มีประสิทธิภาพในการค้นหาโดยรวมดีกว่า เนื่องจากเจอคำตอบทั้งหมด 26 ฟังก์ชัน จากทั้งหมด 27 ฟังก์ชัน และยังใช้เวลาในการค้นหาคำตอบได้น้อยกว่าอัลกอริทึมอื่นๆ แต่เมื่อสังเกตตารางการทดลองแล้วจะเห็นว่า MSPSO ให้ผลในการค้นหาใกล้เคียงกับ CMPSO คือจำนวนฟังก์ชันที่พบคำตอบเท่ากันแต่ MSPSO จะใช้เวลาน้อยกว่า ส่วน GBPSO หาคำตอบเจอน้อยกว่า 1 ฟังก์ชัน(Zakharov) แต่มีเวลาในการค้นหาที่ใกล้เคียง แต่เมื่อลดจำนวนอนุภาคลงเหลือ 30 อนุภาค ซึ่งเป็นค่าที่ใช้ในงานวิจัยของ GBPSO ดังตารางที่ 5.3 - 5.4 ในบทที่ 5 หัวข้อ 5.4 จะเห็นว่า GBPSO สามารถค้นหาคำตอบได้เร็วกว่า GBPSO 90 อนุภาค แต่จะมีบางสมการที่ GBPSO 30 อนุภาคใช้เวลาในการค้นหานานกว่าจะเห็นได้ชัดดังเช่นในฟังก์ชัน Rosenbrock และ Schwefel โดยเฉพาะในฟังก์ชัน Schwefel โดยอาจจะเนื่องมาจากตำแหน่งของคำตอบไม่ได้อยู่ที่ 0 ในทุกมิติจึงทำให้ไม่สามารถหาคำตอบได้ ในขณะที่ GBPSO 90 อนุภาคสามารถหาคำตอบในฟังก์ชันนี้ได้เนื่องจากมีจำนวนอนุภาคที่มากกว่าทำให้มีความหลากหลายของอนุภาคที่มากกว่าส่งผลให้มีโอกาสในการค้นหาเจอคำตอบได้เร็วกว่านั่นเอง และเมื่อทำการเลื่อนตำแหน่งของคำตอบออกจากตำแหน่ง (0,0) ดังตารางที่ 5.3 และ 5.4 ในภาคผนวก ง. หัวข้อ 5.2 จะเห็นว่าทั้ง GBPSO 30 อนุภาค และ 90 อนุภาค นี้เมื่อทำการเลื่อนตำแหน่งแบบสุ่ม(ง.4)โดยที่คำตอบในแต่ละมิติมีค่าที่ไม่เท่ากันจะไม่สามารถหาคำ

คำตอบในฟังก์ชันทั้งสองนี้เจอ แต่ในกรณีปัญหาอื่นๆจะเห็นว่าแค่ 30 อนุภาคก็สามารถหาค่าคำตอบของปัญหาได้แล้วและใช้เวลาในการค้นหาน้อยกว่า

โดยสรุปสำหรับฟังก์ชันที่ยาก มีหลายมิติ และคำตอบในแต่ละมิติไม่ได้อยู่ที่ตำแหน่ง (0,0) และอยู่คนละตำแหน่งกัน วิธีที่นำเสนอ MSPSO จะพบคำตอบ ในขณะที่วิธีอื่นไม่พบคำตอบ ในกรณีที่ไม่ได้มีการshift คำตอบวิธีที่นำเสนอก็พบได้เร็วกว่าวิธีอื่น

6.2 ข้อเสนอแนะ

MSPSO เป็นอัลกอริทึมที่พัฒนามาเพื่อแก้ปัญหาการติด Local Optima ของ PSO ดังนั้นการทำงานหลักๆของ MSPSO ก็จะใช้แนวคิดในการแก้ปัญหาของ GBPSO และ MPSOSM เป็นหลัก เนื่องจาก GBPSO เป็นกลุ่มเดียวและมีการค้นหาที่เร็ว ขณะที่นำ MPSOSM มาพัฒนาและปรับปรุงคือการกลายพันธ์ตำแหน่งของอนุภาคและGBEST และการใช้ GBEST จากกลุ่มข้างเคียง นอกจากนี้ยังนำการเพิ่มลดจำนวนกลุ่มมาช่วยเพิ่มความสามารถในการค้นหาแบบ Global Search และ Local Search ซึ่งทำให้ MSPSO จะเจอค่าคำตอบดีกว่าวิธีอื่น แต่ก็ยังคงมีบางสมการที่ไม่เจอคำตอบหรือไม่หลุดจากการติด Local Optima และมีบางสมการที่ใช้เวลาในการค้นหานานกว่าวิธี PSO เนื่องจากมีการกระจายกลุ่มเป็นหลายอนุภาคทำให้ Evaluation Call เป็น 3 เท่าของ 1 กลุ่ม นอกจากนี้เมื่อมีการทดลองเปลี่ยนตำแหน่งของค่าคำตอบออกจากตำแหน่ง (0,0) จะเห็นว่าวิธี MSPSO นี้ไม่สามารถหาค่าตอบได้ในหลายสมการ แม้มีหลายสมการที่ใช้เวลาในการค้นหาน้อยลงแต่ก็มีอีกหลายสมการที่เจอก็ใช้เวลาในการค้นหานานขึ้น จึงต้องทำการค้นคว้าเพิ่มเติมเพื่อปรับปรุงการกลายพันธ์ และการยุบกระจายกลุ่มให้เหมาะสมกับทุกๆปัญหา เพื่อแก้ไขข้อบกพร่องให้สามารถเจอค่าคำตอบได้ในทุกปัญหาและใช้เวลาในการค้นหาน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] R. C. Eberhart, and J. Kennedy, "Particle Swarm Optimization," IEEE International Conference on Neural Networks, pp. 1942-1948, 1995..
- [2] R. C. Eberhart, and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.
- [3] B. Al-Kazemi, and C. K. Mohan, "Multi-phase discrete particle swarm optimization," In Proc. of 4th Int. Workshop on Frontiers on Evolut.Alg., Research Triangle Park, NC, 2002.
- [4] S. Baskar, and P. N. Suganthan, "A novel concurrent particle swarm optimization," In Proc. of Cong. on Evolut. Comput., Portland, OR, pp. 792-796, 2004.
- [5] C. Li, Q. Zhang, Y. Liu, and L. Kang, "A Fast Particle Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy," Proceedings of the 2nd international conference Advances in computation and Intelligence 2007, Wuhan, China, pp. 334-343.
- [6] K. Chengkhuntod, B. Kruatrachue, and K. Siriboon, "A Modified Multi-Swarm Optimization with Interchange GBEST and Particle Redistribution," Proceedings of The 5th International Electrical Engineering Congress (iEECON), Pattaya, Thailand, pp.1-4, 2017.
- [7] K. Chengkhuntod, B. Kruatrachue, and K. Siriboon, "Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm," Computer Science and Engineering Conference (ICSEC) 2017 21st International, pp. 1-4, 2017.
- [8] V. Cheypoca, K. Siriboon, and B. Kruatrachue, "The use of Global Best position in rerun of particle swarm optimization," Proceedings of the 4th International Conference on Engineering Applied Sciences and technology, Phuket, Thailand, pp.610-613, 2018.
- [9] L. Poempool, B. Kruatrachue, and K. Siriboon, "Combine Multi Particle Swarm in Supporting Trapping in Local Optima," Proceedings of the 4th International Conference on Engineering Applied Sciences and technology, Phuket, Thailand, pp.416-419, 2018.
- [10] A. Hedar, "Test Functions for Unconstrained Optimization", [online], Available: http://www-optima.amp.i.kyotou.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm. 2005.
- [11] S. Surjanovic and D. Bingham. (2017, August). "Optimization Test Functions and Datasets", [Online], Available at: <http://www.sfu.ca/~ssurjano/index.html/>
- [12] A. Rahman.(2018). "Global Optimization Methods and Codes", [Online], Available at: <http://www-optima.amp.i.kyoto-u.ac.jp/>
- [13] Wikipedia contributors.(2018, October). "Test functions for optimization", [Online], Available at: https://en.wikipedia.org/wiki/Test_functions_for_optimization
- [14] Wikipedia contributors.(2017, November). "Multi-swarm optimization", [Online], Available at: https://en.wikipedia.org/wiki/Multi-swarm_optimization

- [15] W. Evans, (2014,September)“EXPLOITATION versus EXPLOITATION MINDSETS in DESIGN-DRIVEN ORGANIZATIONS”,[Online], Available at: https://www.slideshare.net/willewans/exploration-versus-exploitation-mindsets-in-designdriven-organizations/46-BALANCING_THE_PORTFOLIO_Design_resource
- [16] C. Li, Q. Zhang, Y. Liu, and L. Kang, “Fast Multi-swarm Optimization with Cauchy Mutation and Crossover operation,” Proceedings of Advances in computation and Intelligence : Second International Symposium,ISICA2007, Wuhan, China, pp. 344-352.
- [17] Wikipedia contributors.(2018,October). “Optimization problem”,[Online], Available at: https://en.wikipedia.org/wiki/Optimization_problem



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

รายละเอียดของฟังก์ชันมาตรฐานที่ใช้ในการทดลอง

วิทยานิพนธ์ฉบับนี้ได้ทำการทดลองวัดประสิทธิภาพของวิธีการต่างๆที่นำมาใช้ในการค้นหาค่าคำตอบที่ดีที่สุดผ่านฟังก์ชันมาตรฐานทั้งหมด 27 ฟังก์ชัน โดยฟังก์ชันที่นำมาทดลองนั้นใช้ค่ามิติที่ 2, 10 และ 100 มิติ ตามแต่ละเงื่อนไขของฟังก์ชันมาตรฐาน ลักษณะของฟังก์ชันมาตรฐานที่นำมาใช้ในการทดลองแบ่งได้ 2 ลักษณะ ดังนี้

1. Unimodal Function คือฟังก์ชันที่บนพื้นที่การค้นหาจะมีจุด Global optimum และจุด Local optimum คือจุดเดียวกัน ซึ่งดูง่ายต่อการค้นหาแต่เนื่องจากฟังก์ชันลักษณะนี้จะมีพื้นที่ราบจึงมีผลให้อนุภาคเคลื่อนที่เข้าหาคำตอบได้ยาก

2. Multi-modal problem คือฟังก์ชันที่มีความซับซ้อนมากกว่า Unimodal เนื่องจากพื้นที่การค้นหาในรูปแบบความชันที่หลากหลายผสมกัน จึงเป็นสาเหตุให้มีจุด Local optimum ตั้งแต่หนึ่งจุดขึ้นไป แต่มีจุด Global optimum เพียงจุดเดียวเท่านั้น

ทั้งนี้รายละเอียดทั้งหมดของฟังก์ชันมาตรฐานที่ใช้ในการทดลองทั้ง 27 ฟังก์ชันมีดังต่อไปนี้

ก.1 ฟังก์ชัน ACKLEY

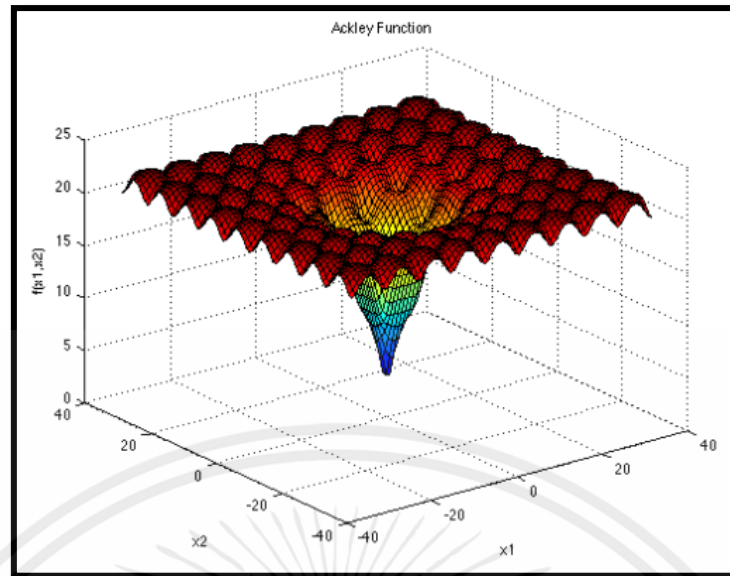
ฟังก์ชัน ACKLEY เป็นฟังก์ชันแบบมี Local Optima มากกว่า 1 จุด ซึ่งจัดอยู่ในแบบมัลติโมดอล มีลักษณะเป็นหลุมลึก ดังรูป ก.1 และสมการของฟังก์ชันจะแสดงดังสมการที่ ก.1

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

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-30, 30]$

ค่า Global Optima คือ $x^* = (0, \dots, 0)$, $f(x^*) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 ฟังก์ชัน ACKLEY

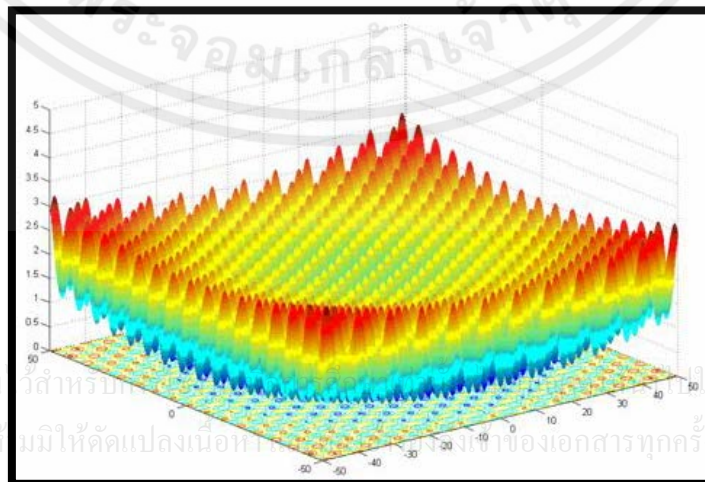
ก.2 ฟังก์ชัน GRIEWANK

ฟังก์ชัน GRIEWANK เป็นฟังก์ชันแบบมัลติโมดอลที่มีความซับซ้อนเนื่องจากมี Local Optima หลายจุดและกระจายอย่างสม่ำเสมอ จะเห็นได้จากรูป ก.2 และสมการของฟังก์ชันคือ สมการ ก.2

$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1 \quad (\text{ก.2})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-300, 300]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.2 ฟังก์ชัน GRIEWANK

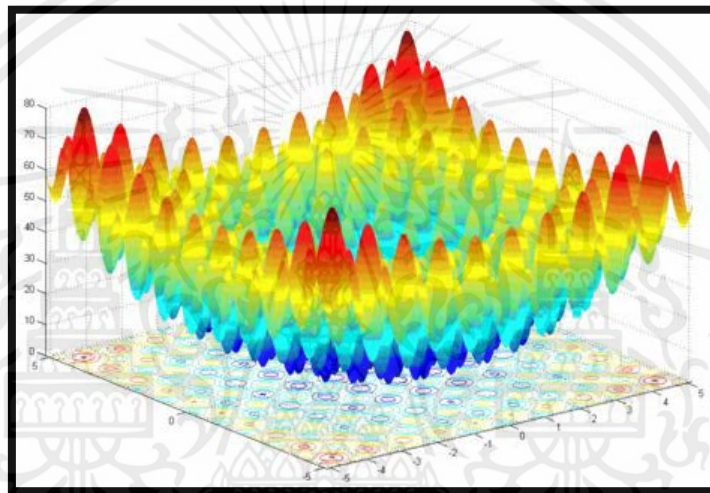
ก.3 ฟังก์ชัน RASTRIGIN

ฟังก์ชัน RASTRIGIN เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.3 และสมการของฟังก์ชันคือสมการ ก.3

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (\text{ก.3})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-5.12, 5.12]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.3 ฟังก์ชัน RASTRIGIN

ก.4 ฟังก์ชัน ROSENBROCK

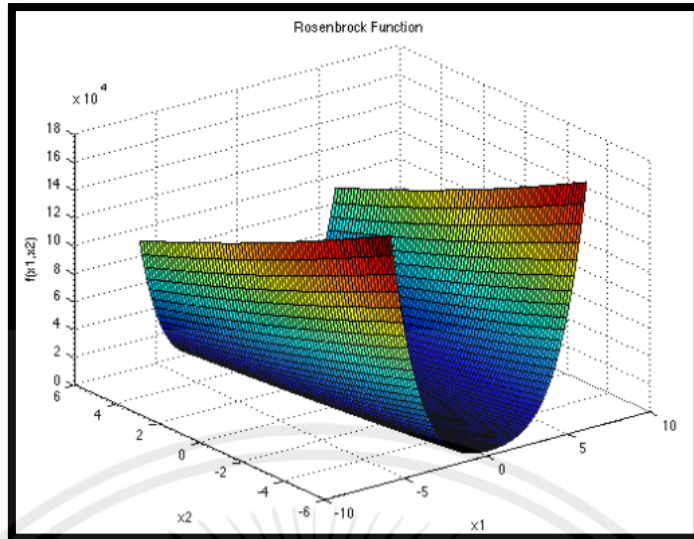
ฟังก์ชัน ROSENBROCK เป็นฟังก์ชันยูนิโมดอล ที่มีรูปแบบ HYPERBOLIC ที่ค่าคำตอบจะอยู่ในช่วงที่แคบๆ ซึ่งดูเหมือนจะเจอค่าคำตอบได้ง่ายแต่การลู่เข้าหาค่าคำตอบที่ดีที่สุดนั้นยาก โดยรูปของฟังก์ชันคือรูป ก.4 และสมการของฟังก์ชันคือสมการ ก.4

$$f(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (\text{ก.4})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-2.048, 2.048]^n$

ค่า Global Optima คือ $x^* = (1, \dots, 1), f(x^*) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องเรียนเท่านั้น ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 ฟังก์ชัน ROSEN BROCK

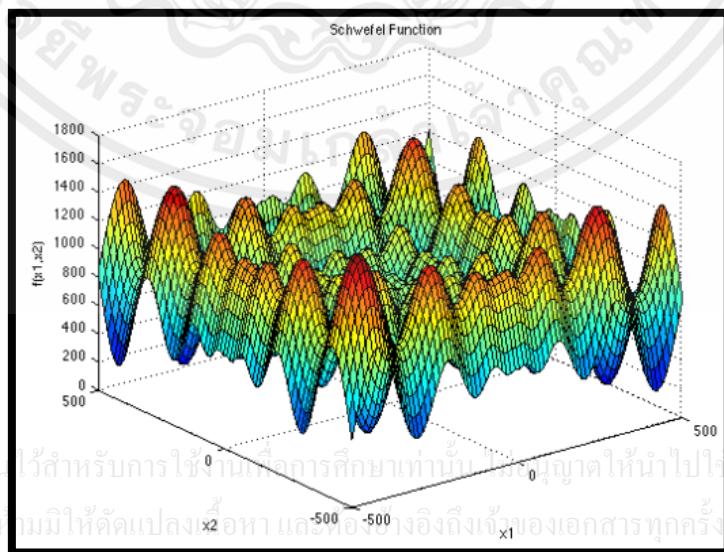
ก.5 ฟังก์ชัน SCHWEFEL

ฟังก์ชัน SCHWEFEL เป็นฟังก์ชันแบบมัลติโมดอล ที่มีความซับซ้อนและมีหลาย Local Optima โดยรูปของฟังก์ชันคือรูป ก.5 และสมการของฟังก์ชัน คือสมการ ก.5

$$f(x) = 418.9829 \times n + \sum_{i=1}^n (x_i \times \sin(\sqrt{|x_i|})) \quad (\text{ก.5})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-500, 500]$

ค่า Global Optima คือ $x^* = (-420.96, \dots, -420.96), f(x^*) = 0$



รูปที่ ก.5 ฟังก์ชัน SCHWEFEL

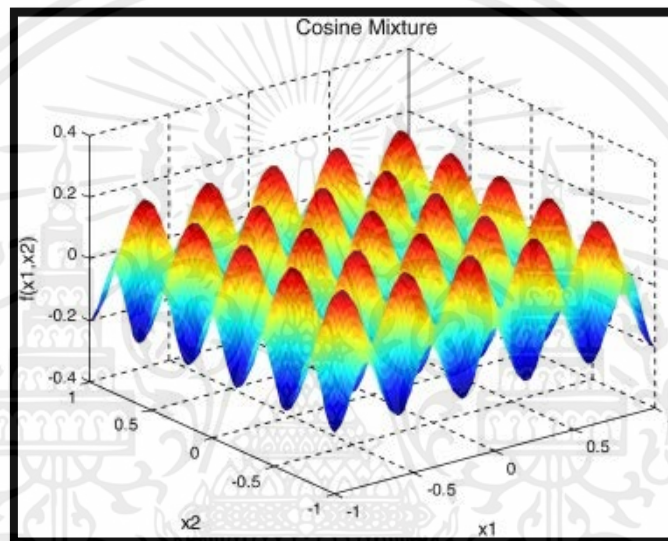
ก.6 ฟังก์ชัน COSINE MIXTURE

ฟังก์ชัน COSINE MIXTURE เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.6 และสมการของฟังก์ชันคือสมการ ก.6

$$f(x) = -0.1 \times \sum_{i=0}^n \cos(5\pi x_i) + \sum_{i=0}^n x^2 + 0.1n \quad (\text{ก.6})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-1,1]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.6 ฟังก์ชัน COSINE MIXTURE

ก.7 ฟังก์ชัน EXPONENTIAL

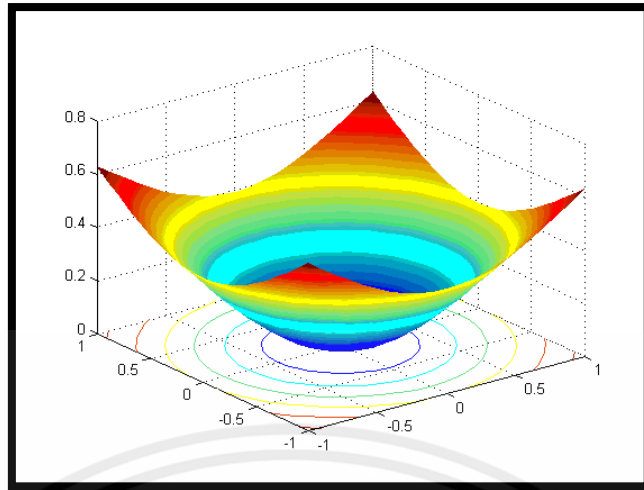
ฟังก์ชัน EXPONENTIAL เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.7 และสมการของฟังก์ชันคือสมการ ก.7

$$f(x) = -\exp(-0.5 \sum_{i=1}^n x_i^2) + 1 \quad (\text{ก.7})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-1,1]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 ฟังก์ชัน EXPONENTIAL

ก.8 ฟังก์ชัน LEVY

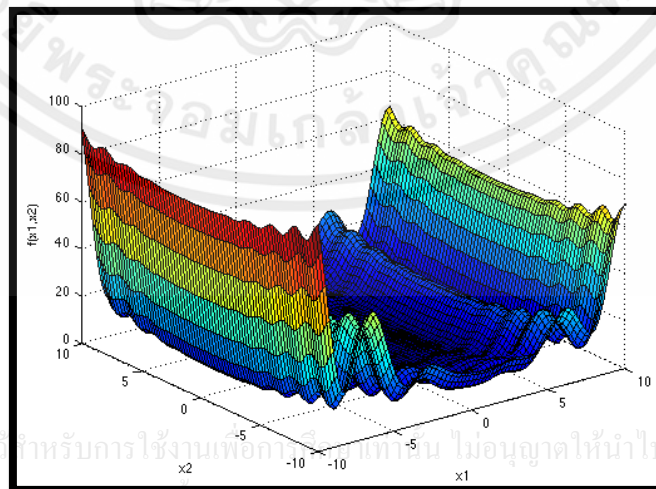
ฟังก์ชัน LEVY เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.8 และสมการของฟังก์ชันคือสมการ ก.8 สำหรับฟังก์ชันนี้ได้ลบค่าด้วย 10^{-30} เพราะถ้าแทนค่า 1 ลงไปในสมการนี้แล้วคำตอบที่ได้ประมาณ 10^{-30} ดังนั้น ถ้าหากลบด้วย จะได้ค่าเท่ากับ 0

$$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)] \quad (\text{ก.8})$$

$$w_i = 1 + \frac{x_i - 1}{4}, \text{ for all } i = 1, \dots, d$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-10, 10]$

ค่า Global Optima คือ $x^* = (1, \dots, 1), f(x^*) = 0$



รูปที่ ก.8 ฟังก์ชัน LEVY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกสิ่งนี้ไป และต้องอ้างอิงถึงเจ้าของเอกสารในทุกครั้งที่มีการนำไปใช้

ก.9 ฟังก์ชัน MICHALEWICZ

ฟังก์ชัน MICHALEWICZ เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.9 และสมการของฟังก์ชันคือสมการ ก.9

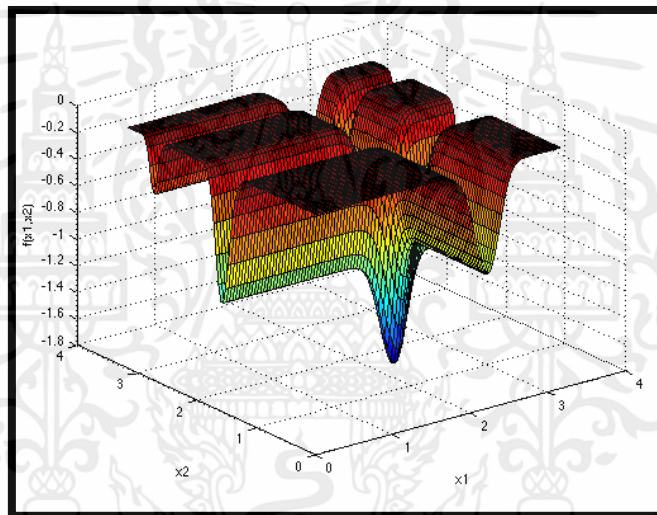
$$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad (\text{ก.9})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [0, \pi]^n$

ค่า Global Optima คือ $d = 2 : f(x^*) = -1.8013, x^* = (2.20, 1.57)$

$$d = 5 : f(x^*) = -4.687658$$

$$d = 10 : f(x^*) = -9.66015$$



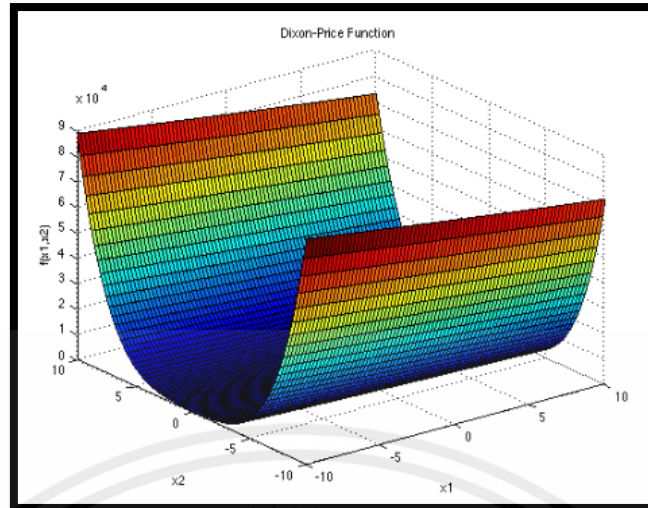
รูปที่ ก.9 ฟังก์ชัน MICHALEWICZ

ก.10 ฟังก์ชัน DIXON-PRICE

ฟังก์ชัน DIXON-PRICE เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.10 และสมการของฟังก์ชันคือสมการ ก.10

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2 \quad (\text{ก.10})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่เป็นประโยชน์เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-10, 10]^n$
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่ข้อมูลอันถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 ค่า Global Optima คือ $f(x^*) = 0$



รูปที่ ก.10 ฟังก์ชัน DIXON-PRICE

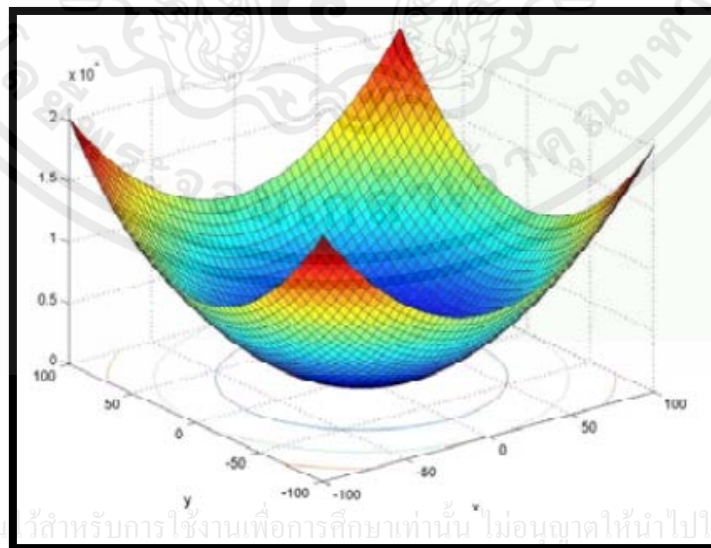
ก.11 ฟังก์ชัน STEP

ฟังก์ชัน STEP เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.11 และสมการของฟังก์ชันคือสมการ ก.11

$$f(x) = \sum_{i=0}^{n-1} (|x_i| + 0.5)^2 \quad (\text{ก.11})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-100, 100]^n$

ค่า Global Optima คือ $x^* = (0.5, \dots, 0.5), f(x^*) = 0$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหามาใช้โดยไม่ได้รับอนุญาตของเอกสารทุกครั้งที่มีกรนำไปใช้

รูปที่ ก.11 ฟังก์ชัน STEP

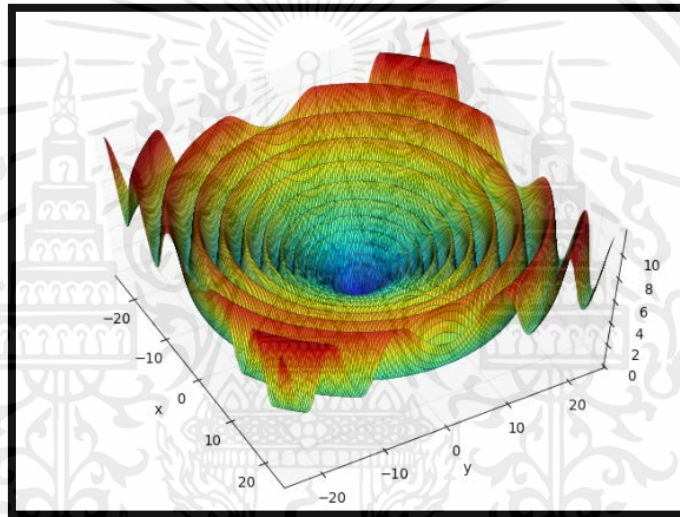
ก.12 ฟังก์ชัน SCHAFFER

ฟังก์ชัน SCHAFFER เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.12 และสมการของฟังก์ชันคือสมการ ก.12

$$f(x) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1.0 + 0.001 \times (x^2 + y^2))^2} \quad (\text{ก.12})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-100, 100]$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.12 ฟังก์ชัน SCHAFFER

ก.13 ฟังก์ชัน HOLDER

ฟังก์ชัน HOLDER เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.13 และสมการของฟังก์ชันคือสมการ ก.13

$$f(x) = \left| \sin(x_1) \cos(x_2) e^{\left(1 - \sqrt{\frac{x_1^2 + x_2^2}{\pi}}\right)} \right| \quad (\text{ก.13})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

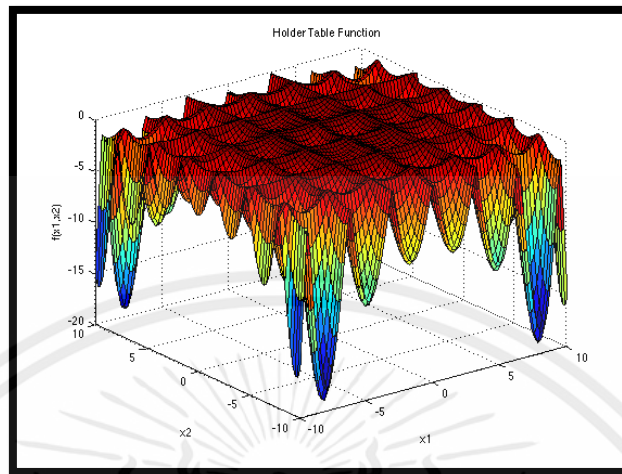
ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-10, 10]$
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Global Optima คือ $x^* = (8.05502, 9.66459), f(x^*) = -19.2085$

$x^* = (8.05502, -9.66459), f(x) = -19.2085$

$$x^* = (-8.05502, 9.66459), f(x) = -19.2085$$

$$x^* = (-8.05502, -9.66459), f(x) = -19.2085$$



รูปที่ ก.13 ฟังก์ชัน HOLDER

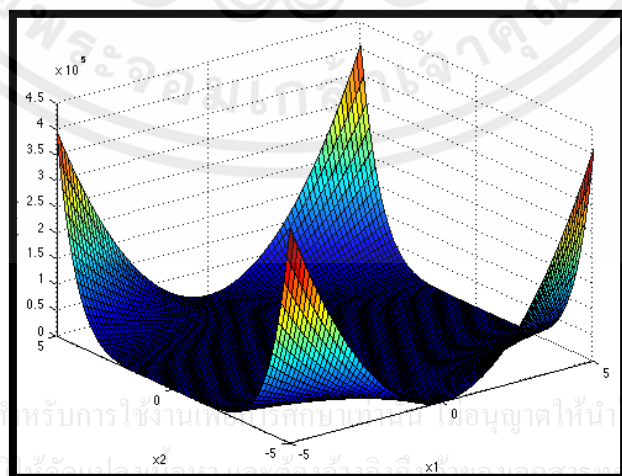
ก.14 ฟังก์ชัน BEALE

ฟังก์ชัน BEALE เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.14 และสมการของฟังก์ชันคือสมการ ก.14

$$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2 \quad (\text{ก.14})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-4.5, 4.5]$

ค่า Global Optima คือ $x^* = (3, 0.5), f(x^*) = 0$



รูปที่ ก.14 ฟังก์ชัน BEALE

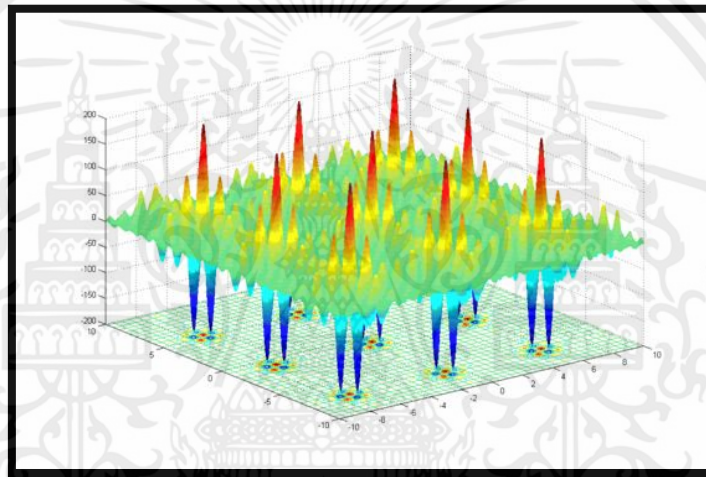
ก.15 ฟังก์ชัน SHUBERT

ฟังก์ชัน SHUBERT เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.15 และสมการของฟังก์ชันคือสมการ ก.15

$$f(x) = \sum_{i=1}^5 i \cos((i+1)x_1 + i) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right) \quad (\text{ก.15})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-10, 10]$

ค่า Global Optima คือ $f(x^*) = -186.7309$



รูปที่ ก.15 ฟังก์ชัน SHUBERT

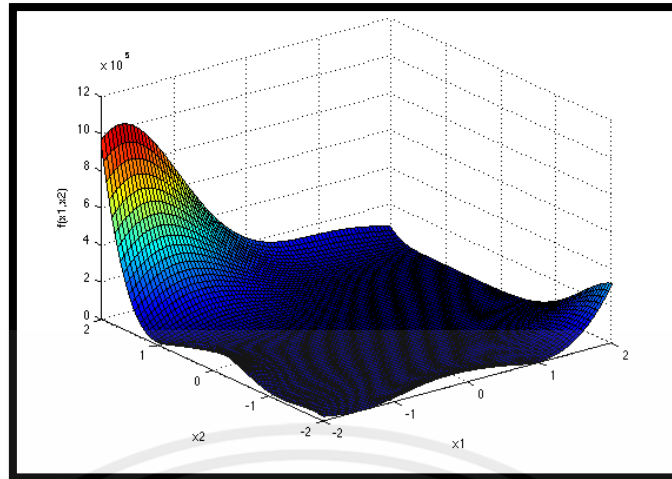
ก.16 ฟังก์ชัน GOLDSTEIN-PRICE

ฟังก์ชัน GOLDSTEIN-PRICE เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.16 และสมการของฟังก์ชันคือสมการ ก.16

$$f(x^*) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \quad (\text{ก.16})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-2, 2]$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ค่า Global Optima คือ $x^* = (0, 1), f(x^*) = 3$
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.16 ฟังก์ชัน GOLDSTEIN-PRICE

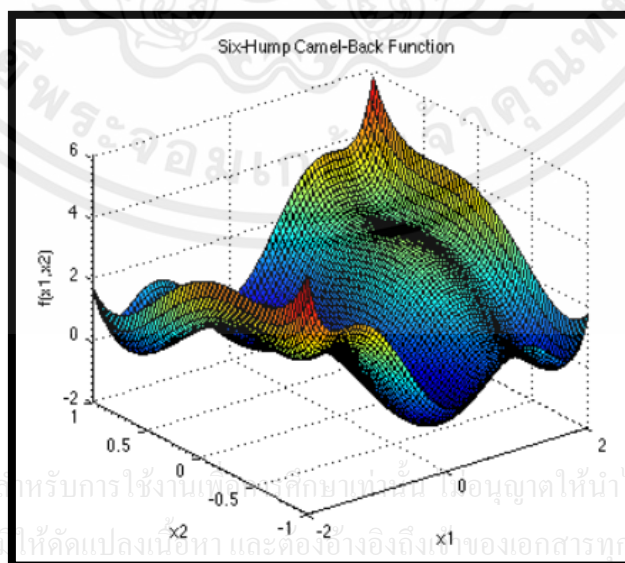
ก.17 ฟังก์ชัน SIX-HUMP CAMEL

ฟังก์ชัน SIX-HUMP CAMEL เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.17 และสมการของฟังก์ชันคือสมการ ก.17

$$f(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2) \quad (\text{ก.17})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-2,2]$

ค่า Global Optima คือ $f(x^*) = 0$



รูปที่ ก.17 ฟังก์ชัน SIX-HUMP CAMEL

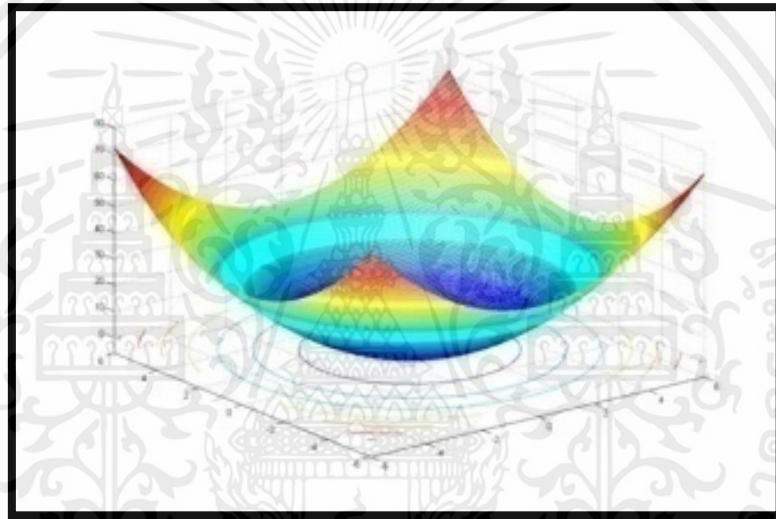
ก.18 ฟังก์ชัน SPHERE

ฟังก์ชัน SPHERE เป็นฟังก์ชันยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.18 และสมการของฟังก์ชันคือสมการ ก.18

$$f(x) = \sum_{i=1}^n x_i^2 \quad (\text{ก.18})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-5.12, 5.12]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.18 ฟังก์ชัน SPHERE

ก.19 ฟังก์ชัน PARALLEL ELLIPSOID

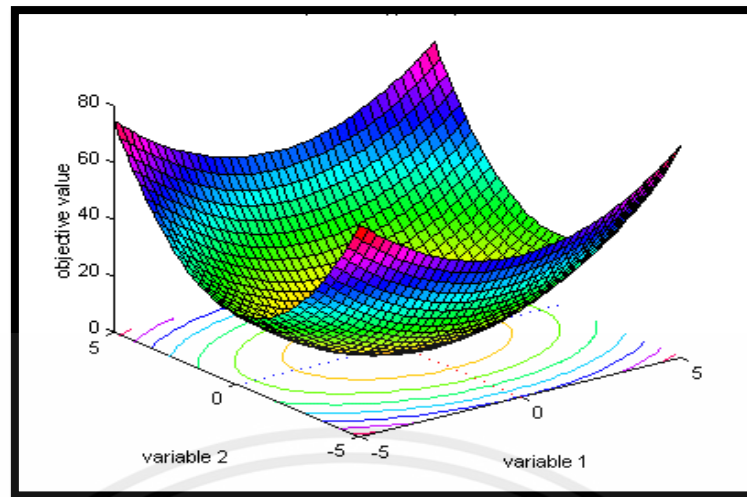
ฟังก์ชัน PARALLEL ELLIPSOID เป็นฟังก์ชันแบบยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.19 และสมการของฟังก์ชันคือสมการ ก.19

$$f(x) = \sum_{i=1}^n (i \times x_i^2) \quad (\text{ก.19})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-5.12, 5.12]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาย่างเป็น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.19 ฟังก์ชัน PARALLEL ELLIPSOID

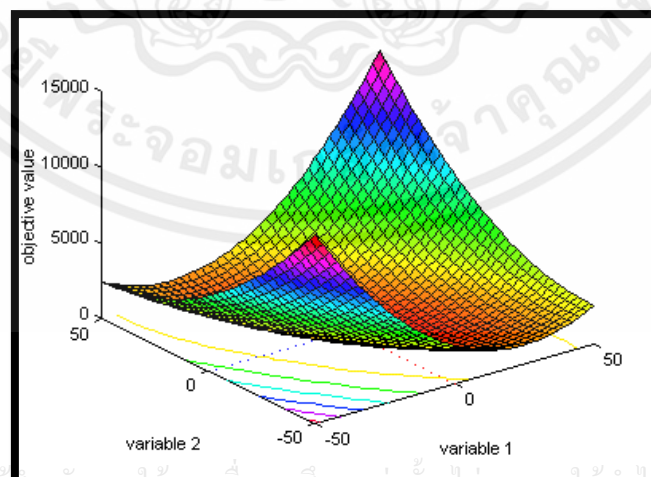
ก.20 ฟังก์ชัน ROTATED ELLIPSOID

ฟังก์ชัน ROTATED ELLIPSOID เป็นฟังก์ชันแบบยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.20 และสมการของฟังก์ชันคือสมการ ก.20

$$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j^2) \quad (\text{ก.20})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-65.536, 65.536]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.20 ฟังก์ชัน ROTATED ELLIPSOID

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาตจากศูนย์ฯ ทุกครั้งที่มีการนำไปใช้

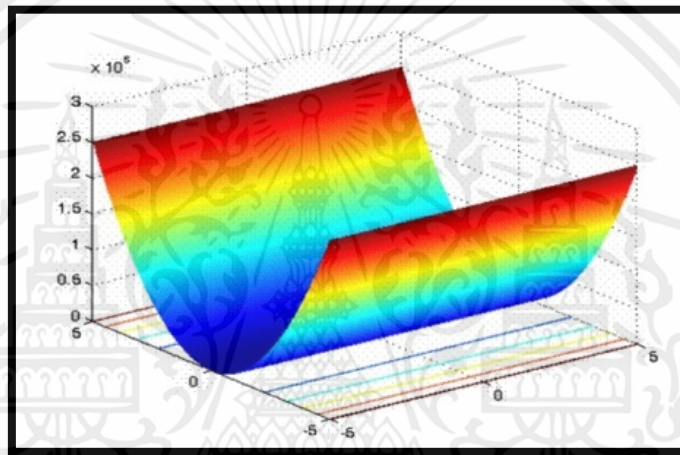
ก.21 ฟังก์ชัน CIGAR

ฟังก์ชัน CIGAR เป็นฟังก์ชันยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.21 และสมการของฟังก์ชันคือสมการ ก.21

$$f(x) = x_0^2 + 10^6 \sum_{i=1}^N x_i^2 \quad (\text{ก.21})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-10,10]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.21 ฟังก์ชัน CIGAR

ก.22 ฟังก์ชัน BROWN

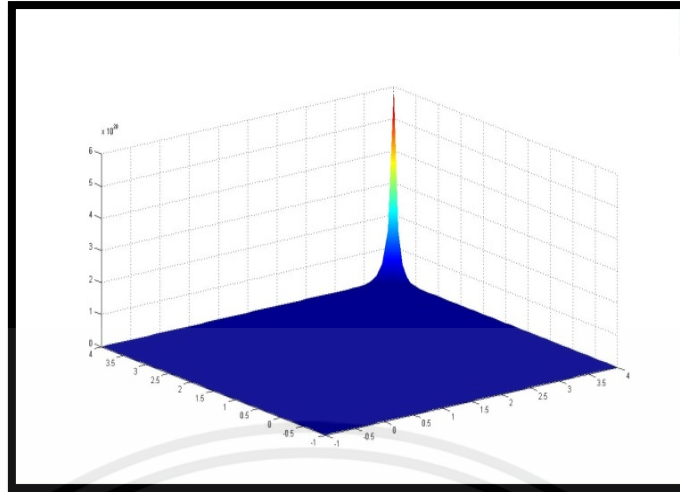
ฟังก์ชัน BROWN เป็นฟังก์ชันยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.22 และสมการของฟังก์ชันคือสมการ ก.22

$$f(x) = \sum_{i=0}^{n-1} [(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}] \quad (\text{ก.22})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-1,4]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.22 ฟังก์ชัน BROWN

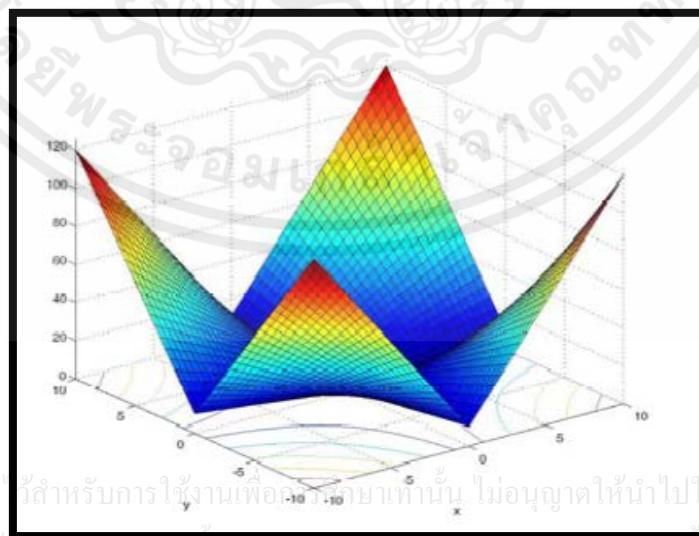
ก.23 ฟังก์ชัน MULTIMOD

ฟังก์ชัน MULTIMOD เป็นฟังก์ชันยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.23 และสมการของฟังก์ชันคือสมการ ก.23

$$f(x) = \sum_{i=1}^n |x_i| \times \prod_{i=1}^n |x_i| \quad (\text{ก.23})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-10,10]$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่บนสื่อใดๆ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.23 ฟังก์ชัน MULTIMOD

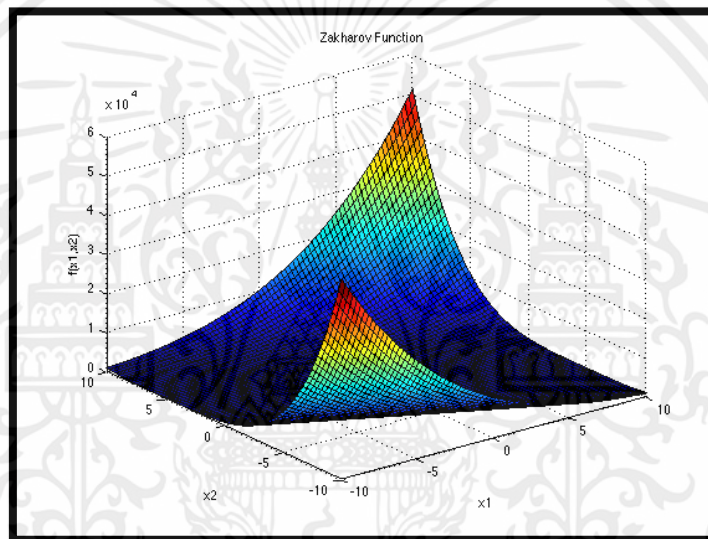
ก.24 ฟังก์ชัน ZAKHAROV

ฟังก์ชัน ZAKHAROV เป็นฟังก์ชันยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.24 และสมการของฟังก์ชันคือสมการ ก.24

$$f(x) = \sum_{i=1}^n x_i^2 + \left[\sum_{i=1}^n \frac{i}{2} x_i \right]^2 + \left[\sum_{i=1}^n \frac{i}{2} x_i \right]^4 \quad (\text{ก.24})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-5, 10]^n$

ค่า Global Optima คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.24 ฟังก์ชัน ZAKHAROV

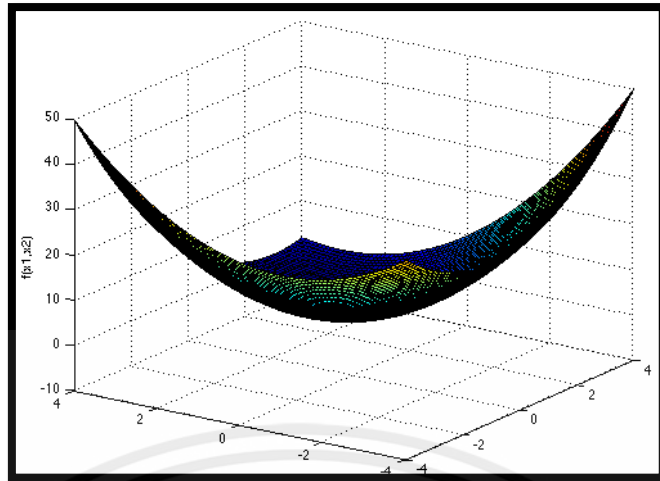
ก.25 ฟังก์ชัน TRID

ฟังก์ชัน TRID เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.25 และสมการของฟังก์ชันคือสมการ ก.25

$$f(x) = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=2}^d x_i x_{i-1} \quad (\text{ก.25})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานวิจัยเท่านั้น การนำฟังก์ชันนี้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกและเผยแพร่ข้อมูลเอกสารทุกครั้งที่มีการนำไปใช้

$$d = 10 : f(x^*) = -200$$



รูปที่ ก.25 ฟังก์ชัน TRID

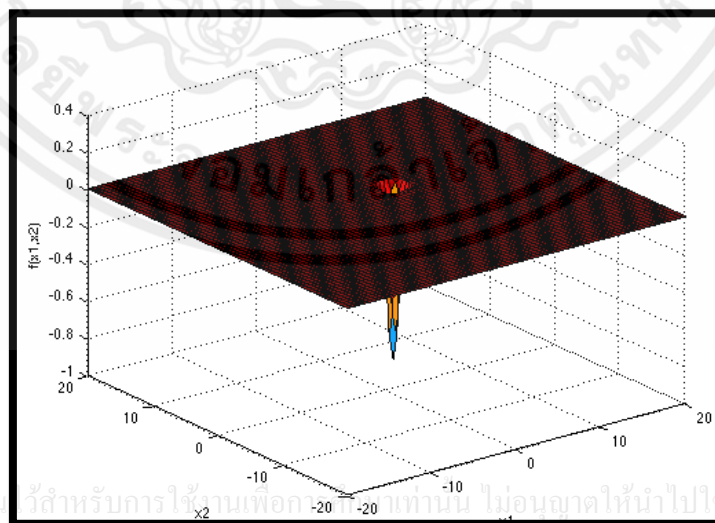
ก.26 ฟังก์ชัน EASOM

ฟังก์ชัน EASOM เป็นฟังก์ชันแบบยูนิโมดอล โดยรูปของฟังก์ชันคือรูป ก.26 และสมการของฟังก์ชันคือสมการ ก.26

$$f(x) = -\cos(x_1)\cos(x_2)\exp\left(-\left(x_1 - \pi\right)^2 - \left(x_2 - \pi\right)^2\right) \quad (\text{ก.26})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-100, 100]$

ค่า Global Optima คือ $x^* = (\pi, \pi), f(x^*) = -1$



รูปที่ ก.26 ฟังก์ชัน EASOM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหานี้ด้วยวิธีการใดๆทั้งที่เป็นของเอกสารทุกครั้งที่มีมีการนำไปใช้

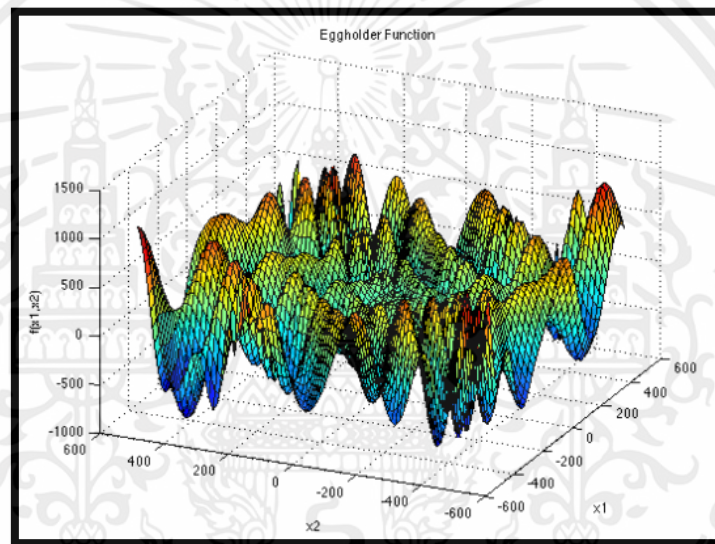
ก.27 ฟังก์ชัน EGGHOLDER

ฟังก์ชัน EGGHOLDER เป็นฟังก์ชันแบบมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.27 และสมการของฟังก์ชันคือสมการ ก.27

$$f(x) = -(x_2 + 47) \sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - x_1 \sin\sqrt{|x_1 - (x_2 + 47)|} \quad (\text{ก.27})$$

ขอบเขตของฟังก์ชันอยู่ในช่วง $x \in [-512, 512]^n$

ค่า Global Optima คือ $x^* = (512, 44.2319)$, $f(x^*) = 0$



รูปที่ ก.27 ฟังก์ชัน EGGHOLDER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ทั้งนี้รายละเอียดทั้งหมดของฟังก์ชันมาตรฐานที่กล่าวในข้างต้นจะสรุปรายละเอียดไว้ใน

ตาราง ก.1

ตารางที่ ก.1 สรุปรายละเอียดพื้นฐานของฟังก์ชันคณิตศาสตร์ที่ใช้ในการทดลอง

ลำดับ	ชื่อฟังก์ชัน	ขอบเขต $[x_{\max}, x_{\min}]$	จุดที่ดีที่สุด	$f(x^*) = 0$ x^*	จำนวน มิติ	คุณสมบัติ
1	ACKLEY	$x \in [-32.768, 32.768]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Multimodal
2	GRIEWANK	$x \in [-300, 300]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Multimodal
3	RASTRIGIN	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Multimodal
4	ROSENBROCK	$x \in [-2.048, 2.048]^n$	$f(x) = 0$	$(1, \dots, 1)$	100	Multimodal
5	SCHWEFEL	$x \in [-500, 500]^n$	$f(x) = 0$	$(420.9687, \dots, 420.9687)$	100	Multimodal
6	COSINE MIXTURE	$x \in [-1, 1]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Multimodal
7	EXPONENTIAL	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Multimodal
8	LEVY	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	$(1, \dots, 1)$	100	Multimodal
9	MICHALEWICZ	$x \in [-65.536, 65.536]^n$	$f(x) = 0$	$(2.20, 1.57)$	100	Multimodal
10	DIXON-PRICE	$x \in [0, P]^n$	$f(x) = 0$	$(0, \dots, 0)$	10	Multimodal
11	STEP	$x \in [-100, 100]^n$	$f(x) = 0$	$(0.5, \dots, 0.5)$	100	Multimodal
12	SCHAFFER	$x \in [-100, 100]^n$	$f(x) = 0$	$(0, \dots, 0)$	2	Multimodal
13	HOLDER	$x \in [-10, 10]^n$	$f(x) = 0$	$(-8.05502, 9.66459)$	2	Multimodal
14	BEALE	$x \in [-4.5, 4.5]^n$	$f(x) = 0$	$(3, 0.5)$	2	Multimodal
15	SHUBERT	$x \in [-10, 10]^n$	$f(x) = 0$	$(0, \dots, 0)$	2	Multimodal
16	GOLDSTEIN-PRICE	$x \in [-2, 2]^n$ * $f(x^*) = 3$	$f(x) = 0$	$(0, -1)$	2	Multimodal
17	SIX-HUMP CAMEL	$x \in [-2, 2]^n$	$f(x) = 0$	$(0.0898, -0.7126)$	2	Multimodal
18	SPHERE	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
19	PARALLEL	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
20	ROTATED	$x \in [-65.536, 65.536]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
21	CIGAR	$x \in [-10, 10]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
22	BROWN	$x \in [-1, 4]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
23	MULTIMOD	$x \in [-10, 10]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
24	ZAKHAROV	$x \in [-5, 10]^n$	$f(x) = 0$	$(0, \dots, 0)$	100	Unimodal
25	TRID	$x \in [-100, 100]^n$	$f(x) = 0$	$(-d^2, d^2)$	10	Multimodal
26	EASOM	$x \in [-100, 100]^n$	$f(x) = 0$	$(0, \dots, 0)$	2	Unimodal
27	EGGHOLDER	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	$(512, 404.2319)$	2	Multimodal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





ภาคผนวก ข

ผลงานวิจัยที่ได้รับการตีพิมพ์

Lukkana Poempool, Boontee Kruatrachue, and Kittawan Siriboon, “Combine Multi Particle Swarm in Supporting Trapping in Local Optima,” the 4th International Conference on Engineering Applied Sciences and technology, 4-7 July 2018, Phuket, Thailand, pp.416-419.




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Conference Proceedings

iceast 2018

The 4th International Conference on Engineering,
Applied Sciences and Technology

"Exploring Innovative Solutions
for Smart Society"

Swissôtel Resort Phuket Patong Beach
Phuket, Thailand
July 4-7, 2018
iceast2018@kmitl.ac.th
www.iceast.org



เอกสารนี้เป็นเอกสารลิขสิทธิ์ของงานวิชาการของสถาบันเทคโนโลยีการเกษตรและวิศวกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำเอกสารไปใช้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Combine multi particle swarm in supporting trapping in local optima

Lukkana Poempool, *Boontee Kruatrachue, and Kritawan Siriboon
 Department of Computer Engineering, Faculty of Engineering
 King Mongkut's Institute of Technology Ladkrabang
 Bangkok, Thailand

E-mail: lukkana.po@gmail.com, *Corresponding Author: boontee.kr@kmitl.ac.th, kritawan.si@kmitl.ac.th

Abstract—This paper proposed using multi swarm to lessen trapping in local optima problem of Particle Swarm Optimization (PSO). The use of multiple swarms can increase wider global search at the cost of decrease narrower local search. Hence, the use of multiple swarms alone can't solve the trapping problem. This paper extends the use of multi swarm by merging all the swarm into single group to enhanced local search when needed. This can increase the local search ability of multi swarm and increase the chance of the trapped swarm to move to other nearby local optima. The proposed method is compared to similarly comparable modified PSO and other similarly multi particle swarm without merging using 26 benchmark functions.

Keywords—Particle Swarm Optimization; PSO; multiple swarm; merging

I. INTRODUCTION

The use of multiple swarm to improve trapping in local optima problem has been widely used by many researcher [3], [4]. By divide swarm into multiple groups with their own particles and best position of the group (GBEST), the entire swarm can search more wildly with multiple group swarms exploring toward their own GBEST. But, since there is less number of particles in each group, the local search performance around multiple GBEST is reduce. Also, multiple swarms can exchange their search information through GBEST and particle positions which can sometime cause the entire groups to search closer. This tradeoff between global and local search and exchange of information can sometimes results in better solution but further improvement are needed.

In order to take advantage of both global and local search, this paper uses the global search in the beginning of the search and finally use local search at the end to improve the final solution. The time to switch from global search to local search is when all the swarm can't be improved and trapped in the same local optima.

To show the effectiveness of the proposed method, it is tested with 26 benchmark functions and comparing to PSO with the same kind of particle mutation when trapped and similar multiple swarm with fixed number of swarm.

This paper organized into five sections. The first section is this introduction. The second section reviews the related

previous works and their algorithms. The third sections explain the proposed method and its algorithm. The fourth section is the experiment results comparing the proposed and other selected methods. The final section is the conclusion.

II. PREVIOUS WORK

A. Particle Swarm Optimization, PSO

PSO is one of the most popular meta-heuristic algorithms. It was proposed by Kennedy and Eberhart in [1], [2]. It was inspired by swarm behavior of birds to find food source which is applied for solving of optimization problem. This algorithm starts from random initial parameter for a swarm of birds or particles in the search space. After that, Particles will move through search space toward the best encounter position of all particles (GBEST) and the best own position of particles from previous search. Each of the particles is replaced by the velocity and position which updated in each round of the search as follows:

$$\vec{v}'_k = \omega \vec{v}_k + c_1 \text{rand}_1(\vec{p}_k - \vec{x}_k) + c_2 \text{rand}_2(\vec{p}_g - \vec{x}_k) \quad (1)$$

$$\vec{x}'_k = \vec{x}_k + \vec{v}'_k \quad (2)$$

Where \vec{x}'_k and \vec{v}'_k are the current position and velocity of a particle k, \vec{x}_k is the previous position of a particle k, \vec{v}_k is the velocity of a particle k, c_1 and c_2 are acceleration constants, rand_1 and rand_2 are random number between 0 and 1, ω is the initial weight, \vec{p}_k is the best position in history (PBEST) and \vec{p}_g is the best position of the swarm(GBEST).

B. Fast Multi-swarm Optimization with Cauchy Mutation and Crossover operation, FMPSO

FMPSO [4] is a multi-particle swarm with Cauchy mutation and cross over. Each swarm has its own GBEST and moving toward its GBEST. Each iteration, a particle selects its next position from position generated from normal PSO and Cauchy mutation from its current position whichever has the better fitness. Then, each particle in a swarm perform crossover with the best particle of a random swarm before starting the next iteration.

C. A Modified Multi-Swarm Optimization with Interchange GBEST and Particle Redistribution, MPSOIR

Chengkhuntod, Kruatrachue and Siriboon [5] proposed A Modified Multi-Swarm Optimization with Interchange GBEST and Particle Redistribution (MPSOIR). They recommend using position of neighboring GBEST and using method of mutation to change the position of each particle when a single group of swarm traps in the same position for some number of iterations. The use of other swarm GBEST is aim to guide the swarm out of its previous trapping GBEST. Even if both the trap swarm's GBEST and the neighbor swarm's GBEST are near or at the same position the mutation of particle positions can also improve the chance to escape the same optima point.

D. Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm, MPSOSM

Chengkhuntod, Kruatrachue and Siriboon [6] improved their works by randomly use of other swarm GBEST instead off fixing the same neighbor GBEST and reduce the mutation range and mutation probability in "Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm :MPSOSM."

MPSOSM use multi-swarm of PSO. Each swarm search and trap in its local optima. When a swarm is trapped, the particle position of the trap swarm is slightly muted from its current position and the best fitness of the trapped swarm is set to those of random other swarm. This can increase the chance of the trapped swarm to move to other local optima. But, if all the swarms are trapped in the same local optima, each swarm will first perform particle position mutation from its current position followed by mutation of its GBEST. All swarm will search in slightly different area, since each swarm performs mutation to different particle positions and different GBEST. Even though the MPSOSM performs better than normal PSO and some multi swarms PSO, some of the benchmark functions still stuck in local optima.

III. THE PROPOSED ALGORITHM

The proposed algorithm "Combine multi particle swarm" (CMPSO) starts with global search with multiple swarms as of MPSOSM. But when all swarm trap in the same local optima, particle of all swarm are slightly mutated and group into one single swarm. And the GBEST of the merging swarm is from the slightly mutation from the previous trap GBEST. This will increase the local search ability due to more particles searching toward the same GBEST.

The pseudo-code of our algorithm is shown in Algorithm 1. The process of PSO is activated in line 1 to 4 of Algorithm 1. This process will run until swarm struck in local optima exceeding some number of iterations. If all swarm trapped in the same local optima, they will process in line 5 to 10 where all PBESTs and GBESTs are reset and all swarm merge into single swarm and all particles' position is reposition slightly from its current position to increase the chance to find better nearby optima. Only 20% of all particle's attribute are changed using (3) with 50% chance of reversing their sign. Since a swarm move toward GBEST, to avoid trapping at the same

GBEST position, GBEST is firstly reset and reposition all particles in swarms. GBEST from the best reposition particles is then slightly modified GBEST using (4) with 30% chance of attribute change. The GBEST attribute change probability is increase to force wider search. But, if swarm trap in different location it process in line 11 to 15 where swarm particle position is also mute with (3), but GBEST is set to GBEST of other random swarm with different GBEST. The number of swarm still the same, more than one group will initially search to the same GBEST. But since each swarm has their own GBEST, it may change and finally search in different location.

$$GBEST_{id} = \pm(GBEST_{id} \times (0.9 + 0.1 \times rand())) \quad (3)$$

$$X_{id} = \pm(X_{id} \times (0.9 + 0.1 \times rand())) \quad (4)$$

ALGORITHM 1: PSEUDO CODE

- (1) Initial particles positions $(X[], V[], GBEST, PBEST[])$
- (2) While(true):
- (3) Update particles of all swarms using normal PSO []
- (4) Evaluate particles from (3) and update PBEST and GBEST of all swarm
- (5) If all sub-swarm trap in the same local optima:
- (6) reset PBEST of all particles, reset GBESTs of all swarms
- (7) randomly mutates of all particle positions by EQ[3]
- (8) evaluate all particle positions and update its PBEST
- (9) Set GBEST to randomly mutate GBEST of the first swarm positions by EQ[3]
- (10) Set number of swarms to 1 //all swarm will have the same GBEST
- (11) Elive (not all sub-swarm trap in the same local optima):
- (12) reset PBEST of all particles, reset GBEST of trapped swarms
- (13) randomly mutation of all particle positions of trap swarm by EQ[3]
- (14) evaluate all particle positions and update PBEST
- (15) set GBEST of all trap swarm to GBEST of other randomly swarm with different GBEST

IV. EXPERIMENT

A. Benchmark Test Functions

The proposed algorithm is compared the performance of another algorithm on 26 benchmark test functions [7], [8] which are described in Table I as below:

TABLE I. BENCHMARK TEST FUNCTIONS

No.	Name Function	Search space	Best Fitness Value	Dim	Best Position
1	ACKLEY	$[0, 32.768, 32.768]^D$	0	100	$(0, \dots, 0)$
2	GREYWALK	$X \in [-300, 300]^D$	0	100	$(0, \dots, 0)$
3	RASTRIGIN	$X \in [-5.12, 5.12]^D$	0	100	$(0, \dots, 0)$
4	ROSENBRCK	$X \in [-2.048, 2.048]^D$	0	100	$(1, \dots, 1)$
5	SCHWEFEL	$X \in [-500, 500]^D$	0	100	$(420.9687, \dots, 420.9687)$
6	COSINE	$X \in [-1, 1]^D$	0	100	$(0, \dots, 0)$
7	EXPONENTIAL	$X \in [-5.12, 5.12]^D$	0	100	$(0, \dots, 0)$
8	LEVY	$X \in [-8.12, 8.12]^D$	0	100	$(1, \dots, 1)$
9	MECHALWICZ	$X \in [-65.536, 65.536]^D$	0	100	$(2.20, \dots, 1.57)$
10	DEXON-PRICE	$X \in [0, 1]^D$	0	10	$(0, \dots, 0)$
11	STEP	$X \in [-100, 100]^D$	0	100	$(0.5, \dots, 0.5)$
12	SCHAFER	$X \in [-100, 100]^D$	0	2	$(0, \dots, 0)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No.	Name Function	Search space	Best Fitness Value	Dim	Best Position
13	HOLDER	$X \in [-10, 10]^P$	0	2	(-0.05502, ..., 9.66459)
14	BEALE	$X \in [-4.5, 4.5]^P$	0	2	(3, ..., 0.5)
15	SHEPHERD	$X \in [-10, 10]^P$	0	2	(0, ..., 0)
16	GOLDSTEIN	$X \in [-2, 2]^P$	0	2	(0, -1)
17	SKO-BUMP	$X \in [-2, 2]^P$	0	2	(0.0898, ..., 0.7127)
18	TRID	$X \in [-100, 100]^P$	0	10	(-d ² , d ²)
19	SPHERE	$X \in [-5, 12.5, 12]^P$	0	100	(0, ..., 0)
20	PARALLEL	$X \in [-5, 12.5, 12]^P$	0	100	(0, ..., 0)
21	ROFATED	$X \in [-65, 536, 65, 536]^P$	0	100	(0, ..., 0)
22	CHOR	$X \in [-10, 10]^P$	0	100	(0, ..., 0)
23	BROWN	$X \in [-1, 1]^P$	0	100	(0, ..., 0)
24	MULTIMOD	$X \in [-10, 10]^P$	0	100	(0, ..., 0)
25	ZAKHAROV	$X \in [-5, 10]^P$	0	100	(0, ..., 0)
26	EASOM	$X \in [-100, 100]^P$	0	2	(0, ..., 0)

The benchmark test functions are both unimodal and multimodal problems. The unimodal problem is search space has one local optimum, while the local optimum and global optimum are the same points. On the other hand, the problem of multimodal is search space there are many local optima and only one point in the local optimum is the global optimum.

From Table I, Functions 1 - 17 are multimodal test functions and 18 - 26 are unimodal test functions. This twenty-six (26) benchmark test functions were used in this the experiment.

B. Experiment setup

The proposed algorithm is compared with MPSOSM, FMP50 and the modified PSO (RPSO). RPSO is a normal single swarm with particle reposition when trap in local optima in the same way as CMP50 except that there is no GBEST reposition and GBEST is set to the best position of particle after reposition.

For a fair comparison, all algorithms have the same parameter as follow: The dimension of parameter is set to be 100. The value of C_1 and C_2 are set to be 1.496180 and the value of ω is set to be 0.729844. The number of iteration is set to be 1,000,000. The population size of RPSO is set to 90 particles and the population size of MPSOSM and CMP50 have 2 cases: They were separated into 3 groups (each group is set to be 30 particles) and into 2 groups (each group is set to be 45 particles)

Our algorithm ran on a Laptop Notebook with a CPU Intel® core™ i5-5200U @2.20GHz and 4 GB of memory (RAM). Algorithm is written in C++ on Microsoft Visual Studio 2015. The number of experiments of each function is 10 times.

C. Experiment Results

The results of the experiment are shown in Table II. This table shows the performance of related algorithms and the proposed algorithm. Only CMP50 can locate global optima in all test functions. MPSOSM and FMP50 fail to find global optima in four and nine test functions respectively. RPSO miss global optima only one function. The number of evaluation calls of CMP50 (45*2) has 12 out of 26 test functions with

minimum evaluation call (the underline number). RPSO also has 8/26 functions with minimum evaluation call, but the number is closer to those of CMP50. Overall, CMP50 has better performance than RPSO, FMP50 and MPSOSM.

TABLE II THE RESULTS OF EXPERIMENTS OF RPSO, FMP50, MPSOSM AND CMP50

NO	Function	RPSO(90)		FMP50 (30*3)		MPSOSM (30*3)		MPSOSM (45*2)		CMP50 (30*3)		CMP50 (45*2)	
		EC*	BF ^b	EC	BF	EC	BF	EC	BF	EC	BF	EC	BF
1	ASLEY	5,118,730	0	75,948,270	0.24	0.24	0.75,934	0	10,000,000	0	5,944,077	0	2,175,234
2	BEALE	<u>2,000,000</u>	0	1,014,200	0	86,000	0	940,777	0	102,900	0	389,000	
3	BROWN	<u>2,000,000</u>	0	75,948,270	732	1,000,012	0	6,175,000	0	1,000,755	0	5,114,256	
4	CHOR	65,847,750	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	0.22	
5	EASOM	45,033,000	0	75,948,270	14,000	1,000,000	0	10,783,370	1100	<u>24,000,000</u>	0	31,163,301	
6	EASOM	33,1270	0	10,019,000	0.240	1,000,304	0	1,013,201	0	1,010,007	0	<u>0.232</u>	
7	FOLDING	1,200,000	0	75,948,270	1,400	1,000,000	0	6,755,011	0	7,400,000	0	<u>680,000</u>	
8	GAFF	5,117,440	0	75,948,270	1,400	1,000,000	0.24	10,000,000	0.24	1,000,000	0	<u>2,000,000</u>	
9	GOLDSTEIN	100,000	0	<u>100,000</u>	0	100,000	0	100,000	0	10,000	0	47,000	
10	GOLDSTEIN	<u>1,000</u>	0	10,000	0	1,750	0	6,770	0	10,000	0	10,000	
11	HOLDER	10,000	0	10,000	0	10,000	0	10,000	0	<u>10,000</u>	0	10,000	
12	HOVER	10,000	0	10,000	0	10,000	0	10,000	0	10,000	0	10,000	
13	HOVER	10,000	0	<u>10,000</u>	0	10,000	0	10,000	0	10,000	0	10,000	
14	HOVER	10,000	0	10,000	0	10,000	0	10,000	0	10,000	0	10,000	
15	HOVER	10,000	0	10,000	0	10,000	0	10,000	0	10,000	0	10,000	
16	HOVER	10,000	0	10,000	0	10,000	0	10,000	0	10,000	0	10,000	
17	HOVER	10,000	0	10,000	0	10,000	0	10,000	0	10,000	0	10,000	
18	TRID	1,114,000	0	1,000,000	0	1,000,000	0	1,000,000	0	<u>10,000</u>	0	10,000	
19	PARALLEL	1,000,000	0	10,000,000	0	10,000,000	0	10,000,000	0	1,000,000	0	<u>1,000,000</u>	
20	PARALLEL	<u>2,000,000</u>	0	10,000,000	0	10,000,000	0	10,000,000	0	1,000,000	0	1,000,000	
21	ROFATED	<u>2,000,000</u>	0	10,000,000	0	10,000,000	0	10,000,000	0	1,000,000	0	1,000,000	
22	SPHERE	<u>2,000,000</u>	0	10,000,000	0	10,000,000	0	10,000,000	0	1,000,000	0	1,000,000	
23	SPHERE	<u>2,000,000</u>	0	10,000,000	0	10,000,000	0	10,000,000	0	1,000,000	0	1,000,000	
24	MULTIMOD	1,000,000	0	10,000,000	0	10,000,000	0	10,000,000	0	1,000,000	0	<u>1,000,000</u>	
25	ZAKHAROV	<u>2,000,000</u>	0	75,948,270	0.24	1,000,000	0	10,000,000	0.24	10,000,000	0	10,000,000	
26	EASOM	10,000	0	75,948,270	0.24	10,000	0	10,000	0	10,000	0	<u>10,000</u>	

* EC = the number of Evaluation call

^b BF = Best Fitness Value

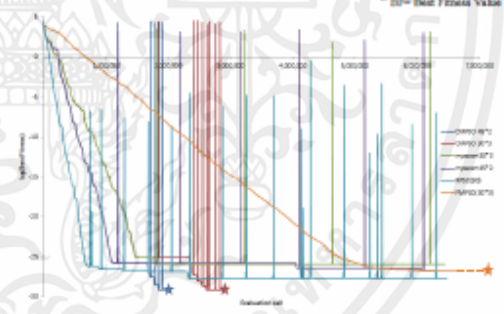


Fig. 1. Convergence rate of all algorithm in function Rosenbrock

Figure 1 shows converging of Rosenbrock function. It shows the GBEST value in log scale on Y-axis and number of evaluation count (time) on X-axis. The vertical jump change of GBEST value is due to the changing of GBEST position when the trapping occurs. Since each algorithm performs differently on the trapping, the convergence to the optimal point is different. RPSO only mute particles for 20% of its dimension so GBEST of RPSO after trapping is still better than those of

CMPSO which has additionally muted for 30%. Since after trapping all swarm are combined into single group, it has better local search and convergence faster than MPSOSM which share the same algorithm until all swarms trap. Only CMPSO and FMPSO reach optima (indicated by star) and CMPSO converge faster than FMPSO but slower than RPSO (RPSO do not reach optima).

V. CONCLUSION

This paper presents Combine multi particle swarm in supporting trapping in local optima by merging all the swarm into single group to enhanced local search. The proposed method not only find global optima in all test function of 100 dimensions but also use less number of evaluation function calls in half of the test functions.

REFERENCES

- [1] R. C. Eberhart, and J. Kennedy, "Particle Swarm Optimization," IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.
- [2] R. C. Eberhart, and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.
- [3] B. Al-Kazemi, and C. K. Mohan, "Multi-phase discrete particle swarm optimization," In Proc. of 4th Int. Workshop on Frontiers on Evolut. Alg., Research Triangle Park, NC, 2002.
- [4] C. Li, Q. Zhang, Y. Liu, and L. Kang, "A Fast Particle Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy," Proceedings of 2nd international conference Advances in computation and Intelligence 2007, Wuhan, China, pp. 334-343.
- [5] K. Chongkumtrod, B. Krautrachua, and K. Siriboon, "A Modified Multi-Swarm Optimization with Interchange GBEST and Particle Redistribution," icsocn, 2017.
- [6] K. Chongkumtrod, B. Krautrachua, and K. Siriboon, "Improving Multi-Swarm by Slightly Mutation Particle and GBEST of Stuck Swarm Along with Randomly Selecting GBEST of other Swarm," icsoc, 2017.
- [7] S. Surjanovic and D. Bingham. (2017, August). Optimization Test Functions and Datasets. [Online] Available at: <https://www.sfu.ca/~ssurjano/optimization.html>
- [8] A. Rahman. (2018). Global Optimization Methods and Codes. [Online] Available at: <http://www-optimia.aup.kyoto-u.ac.jp>



ภาคผนวก ค

การปรับเปลี่ยนค่าพารามิเตอร์ที่เหมาะสมของวิธี MSPSO

ในบทนี้จะทำการทดลองปรับเปลี่ยนค่าพารามิเตอร์ของอัลกอริทึม MSPSO เพื่อหาค่าพารามิเตอร์ที่เหมาะสมที่สุดสำหรับทุกฟังก์ชันที่ใช้ในการทดสอบ โดยค่าพารามิเตอร์ที่นำมาทดสอบมีดังต่อไปนี้

ค.1 การทดลองปรับเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติทั้งหมดที่จะทำการกลายพันธุ์ตำแหน่งของอนุภาคและGBEST

จากการทดลองในบทที่ 5 เปอร์เซ็นต์ของจำนวนมิติทั้งหมดที่จะทำการกลายพันธุ์ตำแหน่งของอนุภาคและGBESTที่ใช้ในการทดลองจะเป็น 20 และ 30 ตามลำดับ โดยการกลายพันธุ์ตำแหน่งนี้จะเกิดขึ้นเมื่ออนุภาคติดอยู่ใน Local Optimum เพื่อเพิ่มโอกาสให้อนุภาคสามารถทำการเคลื่อนที่ไปค้นหาค่าคำตอบในพื้นที่อื่นๆ ซึ่งในวิธีที่เสนอนี้เราได้ทำการปรับเปลี่ยนตำแหน่งการกลายพันธุ์เพียงแค่บางมิติเท่านั้นเนื่องจากก่อนที่อนุภาคจะติดใน Local Optima อนุภาคได้ทำการค้นหาค่าคำตอบมาในระยะเวลาหนึ่งแล้ว ตำแหน่งที่ค่าคำตอบอยู่ควรจะใกล้ๆกับตำแหน่งเดิมมากกว่า ดังนั้นถ้าเราทำการกลายพันธุ์ทุกมิติของอนุภาคและGBESTจะเหมือนกับอนุภาคต้องทำการเริ่มต้นค้นหาค่าคำตอบใหม่ทั้งหมดและอาจทำให้โอกาสในการเจอค่าคำตอบมีน้อยลง ในหัวข้อนี้จะทำการทดลองเพื่อแสดงผลการทดสอบการปรับเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการสุ่มเปลี่ยนตำแหน่งของอนุภาคและGBEST โดยการทดลองนี้จะแบ่งออกเป็นสองการทดลองย่อยคือ

ค.1.1 เปรียบเทียบผลการทดลองว่าจำนวนมิติที่เท่าไรที่เหมาะสมในการสุ่มเปลี่ยนตำแหน่งของอนุภาค โดยจะทำการทดลองเป็น 10%, 20%, 30%, 50%, 70% และ 100% โดยเปอร์เซ็นต์การกลายพันธุ์ตำแหน่งของ GBEST จะเป็น 30% ในทุกกรณี ดังจะแสดงผลตามตารางที่ ค.1 และ ค.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของอนุภาค

NO	Function	10%		20%		30%	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	4,205,392	0	3,463,749	0	2,366,757	0
2	GRIEWANK	365,759	0	343,907	0	346,421	0
3	RASTRIGIN	3,612,852	0	4,981,758	0	3,815,913	0
4	ROSENBROCK	947,327	0	946,389	0	5,278,062	0
5	SCHWEFEL	16,687,196	0	17,834,066	0	34,711,194	0
6	COSINE MIXTURE	574,849	0	553,857	0	539,980	0
7.	EXPONENRIAL	743,441	0	727,883	0	738,275	0
8.	LEVY	1,671,887	0	1,973,691	0	1,569,918	0
9.	MICHALEWICZ	234,990	0	143,230	0	169,289	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0
11.	STEP	128,452	0	121,920	0	131,900	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0
13.	HOLDER	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0
15.	SHUBERT	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0
18	SPHERE	5,432,596	0	5,430,227	0	5,421,850	0
19.	Parallel	5,532,355	0	5,525,623	0	5,525,656	0
20.	Rotated	5,502,375	0	5,503,082	0	5,503,415	0
21.	CIGAR	5,379,552	0	5,379,439	0	5,378,346	0
22.	BROWN	5,397,673	0	5,387,857	0	5,396,337	0
23.	MULTIMOD	14,837,221	0	14,987,233	0	15,312,832	0
24.	ZAKHAROV	85,357,944	0	84,336,016	0	84,485,189	0
25.	TRID	340,649	0	290,386	0	141,741	0
26.	EASOM	18,290	0	18,290	0	18,290	0
27.	EGGHOLDER	90,567,828	0.005078	90,543,127	0.001781	90,515,830	0.000895

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.2 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของอนุภาค
(ต่อ)

NO	Function	50%		70%		100%	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	2,369,656	0	2,717,210	0	1,931,923	0
2	GRIEWANK	345,805	0	345,077	0	343,022	0
3	RASTRIGIN	4,724,498	0	7,842,412	0	3,585,404	0
4	ROSENBROCK	4,028,872	0	1,694,711	0	3,935,051	0
5	SCHWEFEL	90,087,053	355.026	90,063,685	1,539.41	90,053,407	1539.41
6	COSINE MIXTURE	525,574	0	514,285	0	463,961	0
7.	EXPONENTIAL	698,236	0	757,065	0	597,363	0
8.	LEVY	1,482,691	0	2,505,965	0	4,155,179	0
9.	MICHALEWICZ	298,851	0	115,903	0	936,342	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0
11.	STEP	101,799	0	109,855	0	93,552	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0
13.	HOLDER	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0
15.	SHUBERT	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0
18	SPHERE	5,422,037	0	5,423,645	0	5,425,074	0
19.	Parallel	5,527,043	0	5,527,115	0	5,531,076	0
20.	Rotated	5,506,307	0	5,503,581	0	5,502,880	0
21.	CIGAR	5,381,770	0	5,380,225	0	5,379,802	0
22.	BROWN	5,390,091	0	5,394,764	0	5,389,501	0
23.	MULTIMOD	14,580,826	0	15,078,780	0	15,094,539	0
24.	ZAKHAROV	85,795,973	0	84,479,216	0	84,631,190	0
25.	TRID	221,741	0	211,708	0	105,837	0
26.	EASOM	18,290	0	18,290	0	18,290	0
27.	EGGHOLDER	90,470,360	0.000643	90,428,118	0.004101	90,398,601	0.003015

จากตารางที่ ค.1 และ ค.2 จะเห็นว่าการกลายพันธุ์ของอนุภาค 20 เปอร์เซ็นต์ของ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้
จำนวนมิติทั้งหมดของอนุภาคให้ผลใกล้เคียงกับการกลายพันธุ์ของอนุภาคที่ 10 เปอร์เซ็นต์
ไม่ว่ากรณีใดๆทั้งสิ้น
แต่ก็ให้ผลดีกว่าเมื่อเทียบกับการกลายพันธุ์ของอนุภาคด้วยค่าเปอร์เซ็นต์อื่นๆ

ค.1.2 เปรียบเทียบผลการทดลองว่าจำนวนมิติที่เท่าไรที่เหมาะสมในการสุ่มเปลี่ยนตำแหน่งของ GBEST โดยจะทำการทดลอง เป็น 10%, 20%, 30%, 50%, 70% และ 100% โดยเปอร์เซ็นต์การกลายพันธุ์ตำแหน่งของอนุภาคจะเป็น 20% ในทุกกรณี ดังจะแสดงผลตามตารางที่ ค.3 และ ค.4

ตารางที่ ค.3 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของGBEST

NO	Function	10%		20%		30%	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	4,255,888	0	3,721,045	0	3,463,749	0
2	GRIEWANK	346,396	0	344,676	0	343,907	0
3	RASTRIGIN	3,223,459	0	3,457,330	0	4,981,758	0
4	ROSENBROCK	954,438	0	1,061,742	0	946,389	0
5	SCHWEFEL	26,607,904	0	19,489,710	0	17,834,066	0
6	COSINE MIXTURE	591,517	0	583,051	0	553,857	0
7.	EXPONENTIAL	1,025,910	0	569,139	0	727,883	0
8.	LEVY	1,783,343	0	2,823,760	0	1,973,691	0
9.	MICHALEWICZ	112,766	0	112,835	0	143,230	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0
11.	STEP	126,282	0	128,620	0	121,920	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0
13.	HOLDER	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0
15.	SHUBERT	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0
18	SPHERE	5,424,462	0	5,424,566	0	5,430,227	0
19.	Parallel	5,524,846	0	5,528,191	0	5,525,623	0
20.	Rotated	5,502,649	0	5,502,392	0	5,503,082	0
21.	CIGAR	5,381,629	0	5,381,419	0	5,379,439	0
22.	BROWN	5,391,210	0	5,395,646	0	5,387,857	0
23.	MULTIMOD	15,237,789	0	15,150,868	0	14,987,233	0
24.	ZAKHAROV	86,304,025	0	84,224,330	0	84,336,016	0
25.	TRID	298,060	0	238,764	0	290,386	0
26.	EASOM	18,290	0	18,290	0	18,290	0
27.	EGGHOLDER	90,669,352	0.005782	90,595,025	0.006525	90,543,127	0.001781

ตารางที่ ค.4 ตารางแสดงผลการเปลี่ยนเปอร์เซ็นต์ของจำนวนมิติที่จะทำการกลายพันธุ์ของ GBEST

(ต่อ)

NO	Function	50%		70%		100%	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	3,258,769	0	4,130,730	0	2,579,836	0
2	GRIEWANK	348,608	0	344,381	0	368,523	0
3	RASTRIGIN	4,584,406	0	7,088,471	0	3,334,060	0
4	ROSENBROCK	939,698	0	841,517	0	949,130	0
5	SCHWEFEL	33,385,934	0	40,934,281	0	28,383,304	0
6	COSINE MIXTURE	473,960	0	539,539	0	500,618	0
7	EXPONENTIAL	682,028	0	497,442	0	515,020	0
8	LEVY	1,648,297	0	2,932,894	0	1,837,689	0
9	MICHALEWICZ	143,230	0	113,310	0	117,847	0
10	DIXON-PRICE	6,893	0	6,893	0	6,893	0
11	STEP	120,842	0	120,222	0	109,516	0
12	SCHAFFER	20,731	0	20,731	0	20,731	0
13	HOLDER	7,920	0	7,920	0	7,920	0
14	BEALE	32,641	0	32,641	0	32,641	0
15	SHUBERT	6,439	0	6,439	0	6,439	0
16	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0
17	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0
18	SPHERE	5,423,312	0	5,424,130	0	5,422,737	0
19	Parallel	5,525,436	0	5,530,403	0	5,536,282	0
20	Rotated	5,502,155	0	5,502,124	0	5,509,164	0
21	CIGAR	5,380,495	0	5,380,540	0	5,384,098	0
22	BROWN	5,392,579	0	5,395,723	0	5,395,209	0
23	MULTIMOD	14,980,871	0	15,042,149	0	14,858,031	0
24	ZAKHAROV	85,913,139	0	88,852,900	0	84,869,667	0
25	TRID	199,313	0	128,673	0	217,747	0
26	EASOM	18,290	0	18,290	0	18,290	0
27	EGGHOLDER	90,464,114	0.007612	90,414,574	0.001682	90,381,978	0.003302

จากตารางที่ ค.3 และ ค.4 จะเห็นว่าการกลายพันธุ์ของ GBEST 30 เปอร์เซ็นต์ของ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
จำนวนมิติทั้งหมดของ GBEST ให้ผลดีที่สุดเมื่อเทียบกับเปอร์เซ็นต์อื่นๆ
ไม่ว่ากรณีใดๆทั้งสิ้น อย่างไรก็ตามโปรดใช้ความระมัดระวังในการนำข้อมูลไปใช้

ค.2 การทดลองปรับเปลี่ยนจำนวนมิติของฟังก์ชันที่ใช้ในการทดลอง

จากการทดลองในบทที่ 5 จำนวนมิติของฟังก์ชันที่ใช้ในการทดลองจะเป็น 100 มิติ จากตารางที่ 5.2 ในบทที่ 5 จะเห็นว่า จาก 27 ฟังก์ชันจะมีฟังก์ชันที่เป็น 100 มิติ อยู่ 18 ฟังก์ชัน ในหัวข้อนี้จะทำการทดลองเพื่อแสดงผลการทดสอบการปรับเปลี่ยนจำนวนมิติของฟังก์ชันที่ใช้ในการทดลอง โดยทดลองระหว่างจำนวนมิติที่ 30, 60 และ 100 ดังจะแสดงผลตามตารางที่ ค.5

ตารางที่ ค.5 ตารางแสดงผลการทดสอบจำนวนมิติที่ใช้ในการทดลอง

NO	Function	30 Dimension		60 Dimension		100 Dimension	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	1,138,948	0	2,846,532	0	3,463,749	0
2	GRIEWANK	180,965	0	241,900	0	343,907	0
3	RASTRIGIN	371,609	0	1,243,951	0	4,981,758	0
4	ROSENBROCK	144,621	0	554,995	0	946,389	0
5	SCHWEFEL	2,610,823	0	5,267,913	0	17,834,066	0
6	COSINE MIXTURE	93,803	0	269,295	0	553,857	0
7.	EXPONENTIAL	78,147	0	418,965	0	727,883	0
8.	LEVY	185,497	0	622,432	0	1,973,691	0
9.	MICHALEWICZ	143,230	0	143,230	0	143,230	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0
11.	STEP	20,364	0	59,200	0	121,920	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0
13.	HOLDER	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0
15.	SHUBERT	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0
18	SPHERE	1,106,081	0	2,453,651	0	5,430,227	0
19.	Parallel	1,119,646	0	2,460,238	0	5,525,623	0
20.	Rotated	1,102,907	0	2,403,054	0	5,503,082	0
21.	CIGAR	1,124,217	0	2,503,500	0	5,379,439	0
22.	BROWN	1,109,168	0	2,457,118	0	5,387,857	0
23.	MULTIMOD	2,135,886	0	5,905,215	0	14,987,233	0
24.	ZAKHAROV	5,104,512	0	24,314,837	0	84,336,016	0
25.	TRID	245,017	0	329,126	0	290,386	0
26.	EASOM	18,290	0	18,290	0	18,290	0

NO	Function	30 Dimension		60 Dimension		100 Dimension	
		Evaluation	Best	Evaluation	Best	Evaluation	Best
		Call	Fitness	Call	Fitness	Call	Fitness
27.	EGGHOLDER	90,534,885	0.00638	90,553,880	0.007006	90,543,127	0.001781

จากตาราง ค.5 แสดงให้เห็นว่าจำนวนมิติมีผลต่อการค้นหาคำตอบ โดยจำนวนมิติที่น้อยกว่าสามารถหาค่าคำตอบได้เร็วกว่าจำนวนมิติที่มากกว่า

ค.3 การทดลองปรับเปลี่ยนค่า Local Optima Threshold

จากการทดลองในบทที่ 5 เราเลือกใช้ค่า Local Optima Threshold ที่ 100 คือเมื่ออนุภาคไม่เปลี่ยนแปลงครบ 100 รอบจะทำการกลายพันธุ์ตำแหน่งของอนุภาคและ GBEST ตามเงื่อนไขที่กำหนด ในหัวข้อนี้จะทำการทดลองเพื่อแสดงผลการทดสอบการปรับเปลี่ยนค่า Local Optima Threshold โดยทดลองระหว่างค่า Local Optima Threshold ที่ 30, 50, 100 และ 150 รอบ ดังจะแสดงผลตามตารางที่ ค.6

ตารางที่ ค.6 ตารางแสดงผลการทดสอบค่า Local Optima Threshold

NO	Function	Threshold 30		Threshold 50		Threshold 100		Threshold 150	
		Evaluation	Best	Evaluation	Best	Evaluation	Best	Evaluation	Best
		Call	Fitness	Call	Fitness	Call	Fitness	Call	Fitness
1	ACKLEY	1,449,316	0	1,872,889	0	3,463,749	0	4,556,670	0
2	GRIEWANK	327,198	0	340,462	0	343,907	0	379,863	0
3	RASTRIGIN	2,201,441	0	4,558,714	0	4,981,758	0	5,181,935	0
4	ROSENBROCK	1,820,636	0	1,062,660	0	946,389	0	902,060	0
5	SCHWEFEL	31,713,876	0	30,145,155	0	17,834,066	0	35,454,410	0
6	COSINE MIXTURE	488,546	0	497,005	0	553,857	0	548,262	0
7.	EXPONENTIAL	428,437	0	592,867	0	727,883	0	906,618	0
8.	LEVY	1,320,158	0	2,652,951	0	1,973,691	0	5,539,242	0
9.	MICHALEWICZ	45,046	0	369,487	0	143,230	0	250,816	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0	6,893	0
11.	STEP	96,134	0	111,265	0	121,920	0	154,171	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0	20,731	0
13.	HOLDER	7,920	0	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0	32,641	0

NO	Function	Threshold 30		Threshold 50		Threshold 100		Threshold 150	
		Evaluation	Best	Evaluation	Best	Evaluation	Best	Evaluation	Best
		Call	Fitness	Call	Fitness	Call	Fitness	Call	Fitness
15.	SHUBERT	6,439	0	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0	2,003	0
18	SPHERE	5,407,706	0	5,419,970	0	5,430,227	0	5,426,996	0
19.	Parallel	5,502,179	0	5,511,496	0	5,525,623	0	5,529,567	0
20.	Rotated	5,488,216	0	5,493,186	0	5,503,082	0	5,506,805	0
21.	CIGAR	5,374,165	0	5,376,418	0	5,379,439	0	5,384,236	0
22.	BROWN	5,380,600	0	5,386,047	0	5,387,857	0	5,394,766	0
23.	MULTIMOD	48,330,117	0	15,982,810	0	14,987,233	0	14,207,848	0
24.	ZAKHAROV	91,107,580	394.974	90,321,454	36.023	84,336,016	0	84,825,475	0
25.	TRID	167,329	0	153,975	0	290,386	0	320,502	0
26.	EASOM	23,658	0	18,290	0	18,290	0	18,290	0
27.	EGGHOLDER	91,252,984	0.006338	90,958,591	0.004809	90,543,127	0.001781	90,391,867	2.36E-05

จากตารางที่ ค.6 จะเห็นว่า ค่า Local Optima Threshold มีผลต่อการค้นหาค่าคำตอบ เมื่อทำการกำหนดค่า Local Optima Threshold ที่ 30 รอบ และ 50 รอบ จะเห็นว่าเจอคำตอบได้น้อยกว่าเมื่อกำหนดค่า Local Optima Threshold ที่ 100 รอบ และมีจำนวน Evaluation Call มากกว่าในหลายฟังก์ชัน ขณะที่ ค่า Local Optima Threshold ที่ 150 รอบ มีจำนวน Evaluation Call โดยรวมน้อยกว่าในกรณีที่เจอคำตอบ แต่จำนวนฟังก์ชันที่เจอคำตอบมีน้อยกว่าค่า Local Optima Threshold ที่ 100 และ 150 รอบ 2 ค่านี้จะเจอคำตอบฟังก์ชันมากกว่า ถึงแม้บางฟังก์ชันจะใช้ Evaluation Call มากกว่าค่า Local Optima Threshold ที่ 30 และ 50 รอบ

ค.4 การทดลองปรับเปลี่ยนจำนวนกลุ่มที่ใช้ในการกระจายอนุภาค

จากการทดลองในบทที่ 5 ได้กำหนดจำนวนกลุ่มที่ใช้ในการแบ่งกลุ่มอนุภาคเพื่อใช้ในการค้นหาแบบ Multi Group ที่ใช้ในการทดลองคือ 3 กลุ่ม กลุ่มละ 30 อนุภาค ในหัวข้อ ค.4 นี้จะทำการทดลองเพื่อแสดงผลการทดสอบการปรับเปลี่ยนจำนวนกลุ่มที่ใช้ในการแบ่งกลุ่มอนุภาคที่ใช้ในการทดลอง โดยทำการทดสอบระหว่างจำนวน 2 กลุ่ม กลุ่มละ 45 อนุภาค, 3 กลุ่ม กลุ่มละ 30 อนุภาค และ 5 กลุ่ม กลุ่มละ 18 อนุภาค ดังจะแสดงผลตามตารางที่ ค.7

ตารางที่ ค.7 ตารางแสดงผลการทดสอบจำนวนกลุ่มที่ใช้ในการกระจายอนุภาค

NO	Function	18x5		30x3		45x2	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	3,146,190	0	3,463,749	0	2,896,065	0
2	GRIEWANK	343,907	0	343,907	0	343,907	0
3	RASTRIGIN	3,874,863	0	4,981,758	0	4,780,330	0
4	ROSENBROCK	946,389	0	946,389	0	946,389	0
5	SCHWEFEL	29,146,517	0	17,834,066	0	37,027,842	0
6	COSINE MIXTURE	553,857	0	553,857	0	553,857	0
7.	EXPONENTIAL	737,961	0	727,883	0	1,104,448	0
8.	LEVY	2,215,069	0	1,973,691	0	2,579,700	0
9.	MICHALEWICZ	143,230	0	143,230	0	143,230	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0
11.	STEP	121,920	0	121,920	0	121,920	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0
13.	HOLDER	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0
15.	SHUBERT	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN- PRICE	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0
18	SPHERE	5,430,227	0	5,430,227	0	5,430,227	0
19.	Parallel	5,525,623	0	5,525,623	0	5,525,623	0
20.	Rotated	5,503,082	0	5,503,082	0	5,503,082	0
21.	CIGAR	5,379,439	0	5,379,439	0	5,379,439	0
22.	BROWN	5,387,857	0	5,387,857	0	5,387,857	0
23.	MULTIMOD	15,180,440	0	14,987,233	0	23,030,509	0
24.	ZAKHAROV	84,426,842	0	84,336,016	0	84,231,390	0
25.	TRID	312,262	0	290,386	0	256,347	0
26.	EASOM	18,290	0	18,290	0	18,290	0
27.	EGGHOLDER	90,575,098	0.00593	90,543,127	0.001781	90,565,630	0.000737

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 จากตาราง ค.7 จะเห็นว่าจำนวนกลุ่มมีผลต่อการค้นหาค่าตอบในแต่ละฟังก์ชัน จากตารางจะ
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 เห็นว่า เมื่อแบ่งจำนวนกลุ่มเป็น 5 กลุ่ม แต่ละกลุ่มมีเพียง 18 อนุภาค ก็ยังสามารถหาค่าค่าตอบได้แต่

ว่าช้ากว่าเมื่อแบ่งจำนวนกลุ่มเป็น 3 กลุ่ม กลุ่มละ 30 อนุภาค ขณะเดียวกันเมื่อแบ่งจำนวนกลุ่มเหลือเพียง 2 กลุ่ม กลุ่มละ 45 อนุภาค ก็มีหลายปัญหาเช่นกันที่หาค่าคำตอบได้ช้ากว่าเมื่อแบ่งแบบ 3 กลุ่ม ดังนั้นการแบ่งกลุ่มออกเป็น 3 กลุ่ม กลุ่มละ 30 อนุภาค จึงเป็นค่าที่เหมาะสมที่สุดในการทดสอบวิธี MSPSO

ค.5 การทดลองปรับเปลี่ยนจำนวนครั้งในการกระจายอนุภาคทั้งแบบหลายกลุ่มอนุภาค และแบบ 1 กลุ่มอนุภาค

จากการทดลองในบทที่ 5 ได้กำหนดจำนวนการกระจายอนุภาคแบบ 1 กลุ่มอนุภาค (Reposition of Single Group) ที่ 6 ครั้ง และแบบหลายกลุ่มอนุภาค (Reposition of Multi Group) ที่ 4 ครั้ง ในหัวข้อ ค.5 นี้จะทำการทดลองเพื่อแสดงผลการทดสอบการปรับเปลี่ยนจำนวนครั้งที่ใช้ในการกระจายอนุภาคทั้งแบบหลายกลุ่มอนุภาคและแบบ 1 กลุ่มอนุภาค โดยทำการทดสอบระหว่างการ Reposition of Single Group (S) 1 ครั้ง - การ Reposition of Multi Group (M) 1 ครั้ง, S3-M3, S5-M5 และ S6-M4 ดังจะแสดงผลตามตารางที่ ค.8

ตารางที่ ค.8 ตารางแสดงผลการทดสอบการปรับเปลี่ยนจำนวนครั้งในการกระจายอนุภาคแบบหลายกลุ่มอนุภาคและแบบ 1 กลุ่มอนุภาค

NO	Function	S1-M1		S3-M3		S5-M5		S6-M4	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	3,352,561	0	3,690,059	0	4,115,887	0	3,463,749	0
2	GRIEWANK	342,444	0	343,907	0	343,907	0	343,907	0
3	RASTRIGIN	4,427,507	0	4,051,414	0	3,362,368	0	4,981,758	0
4	ROSENBROCK	918,022	0	932,433	0	946,389	0	946,389	0
5	SCHWEFEL	22,912,464	0	37,610,669	0	37,189,415	0	17,834,066	0
6	COSINE MIXTURE	716,224	0	894,369	0	574,253	0	553,857	0
7.	EXPONENTIAL	776,791	0	791,687	0	959,180	0	727,883	0
8.	LEVY	2,001,427	0	3,052,342	0	2,307,077	0	1,973,691	0
9.	MICHALEWICZ	122,249	0	143,230	0	143,230	0	143,230	0
10.	DIXON-PRICE	6,893	0	6,893	0	6,893	0	6,893	0
11.	STEP	125,324	0	121,920	0	121,920	0	121,920	0
12.	SCHAFFER	20,731	0	20,731	0	20,731	0	20,731	0

NO	Function	S1-M1		S3-M3		S5-M5		S6-M4	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
13.	HOLDER	7,920	0	7,920	0	7,920	0	7,920	0
14.	BEALE	32,641	0	32,641	0	32,641	0	32,641	0
15.	SHUBERT	6,439	0	6,439	0	6,439	0	6,439	0
16.	GOLDSTEIN-PRICE	10,760	0	10,760	0	10,760	0	10,760	0
17.	SIX-HUMP CAMEL	2,003	0	2,003	0	2,003	0	2,003	0
18.	SPHERE	5,802,065	0	5,430,227	0	5,430,227	0	5,430,227	0
19.	Parallel	5,528,153	0	5,525,623	0	5,525,623	0	5,525,623	0
20.	Rotated	5,503,127	0	5,503,082	0	5,503,082	0	5,503,082	0
21.	CIGAR	5,390,542	0	5,379,439	0	5,379,439	0	5,379,439	0
22.	BROWN	5,392,946	0	5,387,857	0	5,387,857	0	5,387,857	0
23.	MULTIMOD	18,498,877	0	15,030,130	0	15,278,209	0	14,987,233	0
24.	ZAKHAROV	89,998,904	2.37E-254	89,999,825	5.75E-308	84,197,653	0	84,336,016	0
25.	TRID	485,957	0	204,140	0	303,567	0	290,386	0
26.	EASOM	18,290	0	18,290	0	18,290	0	18,290	0
27.	EGGHOLDER	90,557,944	0.00203	90,554,831	0.000399	90,545,214	0.000811	90,543,127	0.001781

จากตาราง ค.8 จะเห็นว่าเมื่อมีการกำหนดการรีโพลิชันแบบหนึ่งกลุ่มอนุภาคที่น้อยเกินไป อาจทำให้อนุภาคที่กำลังจะเคลื่อนที่ไปเจอค่าคำตอบต้องเสียเวลาในการกระจายอนุภาคเพื่อหาค่าคำตอบใหม่อีกครั้งหนึ่ง จึงควรทำการค้นหาแบบหนึ่งกลุ่มอนุภาคจนแน่ใจแล้วว่าไม่สามารถหาค่าคำตอบได้แล้วถึงค่อยทำการแบ่งกลุ่มอนุภาคออกเป็นหลายกลุ่มเพื่อเพิ่มพื้นที่ในการค้นหาค่าคำตอบและเพิ่มโอกาสในการเจอคำตอบได้เร็วขึ้นนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

การเลื่อนตำแหน่งโดเมนของฟังก์ชันมาตรฐาน

ในบทนี้ทำการทดลองเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดของปัญหาจากตำแหน่งจุด (0,0) ไปเป็นตำแหน่งอื่น หรือการเลื่อนตำแหน่งโดเมนของฟังก์ชันมาตรฐาน เนื่องจากพฤติกรรมการค้นหาคำตอบของ PSO จะทำการเคลื่อนที่ไปหาค่าตอบที่ดีที่สุดจนกว่าค่าที่ดีที่สุดจะไม่ดีขึ้นแล้วหรือครบรอบที่กำหนดไว้ในการค้นหา เนื่องจากวิธีที่นำเสนอนี้มีแนวโน้มที่เหมาะสมกับการหาค่าคำตอบที่อยู่ในตำแหน่ง (0,...,0) จึงได้ทำการทดลองเลื่อนตำแหน่งของคำตอบไม่ให้อยู่ที่ตำแหน่ง (0,...,0) โดยการทำทดสอบมีดังต่อไปนี้

ง.1 การทดลองเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่งอื่นๆเพื่อทดสอบประสิทธิภาพของวิธี MSPSO

จากการทดลองในบทที่ 5 วิธีที่นำเสนอหรือ MSPSO จะเห็นว่าสามารถหาค่าคำตอบที่ดีที่สุดได้ 26 ฟังก์ชัน และใช้เวลาการค้นหาโดยรวมได้เร็วกว่าอัลกอริทึมอื่นๆ ในภาคผนวกนี้จึงได้ทำการเลื่อนตำแหน่งของค่าคำตอบเพื่อทดสอบประสิทธิภาพของอัลกอริทึม MSPSO นี้ โดยตำแหน่งที่เลือกมาทำการทดสอบนั้นมี 2 กรณี คือ กรณีที่เลื่อนตำแหน่งของค่าคำตอบด้วยการบวกค่าคงที่ (0.1, 0.01, 0.00001, 25% ของขอบเขตบนของแต่ละฟังก์ชัน) และกรณีเลื่อนตำแหน่งของค่าคำตอบด้วยการบวกค่าที่ได้จากการสุ่ม (Random Shift) ดังตารางด้านล่าง

ตาราง ง.1 ตารางแสดงผลการค้นหาค่าคำตอบของวิธี MSPSO โดยทำการเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่งอื่นๆ

NO	Function	0.1		0.01		0.00001	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	90,137,466	2.84E-14	90,143,201	2.84E-14	90,190,920	2.84E-14
2	GRIEWANK	1,129,863	0	1,284,684	0	367,051	0
3	RASTRIGIN	6,938,312	0	6,402,999	0	4,423,270	0
4	ROSENBROCK	14,226,005	0	2,211,629	0	3,012,856	0
5	SCHWEFEL	24,415,618	0	32,133,500	0	36,003,198	0
6	COSINE MIXTURE	2,088,899	0	4,573,808	0	1,067,616	0

NO	Function	0.1		0.01		0.00001	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
7.	EXPONENTIAL	90,221,919	2.22E-16	90,259,047	2.22E-16	90,428,090	2.22E-16
8.	LEVY	1,819,074	0	1,791,119	0	3,073,863	0
9.	MICHALEWICZ	105,328	0	1,080,145	0	219,759	0
10.	DIXON-PRICE	103,983	0	90,872,727	0.003142	90,886,001	4.20E-09
11.	STEP	113,490	0	148,212	0	114,113	0
12.	SCHAFFER	17,425	0	397,409	0	19,921	0
13.	HOLDER	6,946	0	5,400	0	7,920	0
14.	BEALE	30,348	0	32,502	0	32,987	0
15.	SHUBERT	6,686	0	5,285	0	6,529	0
16.	GOLDSTEIN-PRICE	11,118	0	11,389	0	12,021	0
17.	SIX-HUMP CAMEL	54	0	6,827	0	2,003	0
18.	SPHERE	869,587	0	954,097	0	987,300	0
19.	Parallel	872,893	0	825,055	0	968,094	0
20.	Rotated	840,375	0	824,415	0	1,035,827	0
21.	CIGAR	788,566	0	1,041,639	0	1,003,716	0
22.	BROWN	899,223	0	824,641	0	948,429	0
23.	MULTIMOD	1,656,827	0	2,100,169	0	1,298,456	0
24.	ZAKHAROV	90,002,520	6.23E-31	90,005,733	4.65E-33	90,006,342	4.70E-39
25.	TRID	165,568	0	70,613	0	324,081	0
26.	EASOM	18,939	0	19,769	0	18,560	0
27.	EGGHOLDER	238,670	0	8,107,867	0	90,527,958	0.005074

ตาราง ง.2 ตารางแสดงผลการค้นหาค่าคำตอบของวิธี MSPSO โดยทำการเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่งอื่นๆ (ต่อ)

NO	Function	Shift 25% of Upper limit		Random shift		MSPSO Original	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	90,113,228	2.84E-14	90,116,970	2.84E-14	3,463,749	0
2	GRIEWANK	2,145,092	0	5,359,618	0	343,907	0
3	RASTRIGIN	19,888,830	0	15,070,312	0	4,981,758	0
4	ROSENBROCK	1,046,760	0	90,087,871	1.48E-29	946,389	0
5	SCHWEFEL	253,220	0	90,063,065	4623.63	17,834,066	0

NO	Function	Shift 25% of Upper limit		Random shift		MSPSO Original	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
6	COSINE MIXTURE	4,077,945	0	1,758,777	0	553,857	0
7	EXPONENTIAL	90,212,490	3.33E-16	90,222,664	3.33E-16	727,883	0
8	LEVY	3,440,608	0	1,663,571	0	1,973,691	0
9	MICHALEWICZ	235,770	0	135,882	0	143,230	0
10	DIXON-PRICE	90,892,363	505.169	90,609,356	0.090251	6,893	0
11	STEP	207,654	0	132,324	0	121,920	0
12	SCHAFFER	31,611	0	17,935	0	20,731	0
13	HOLDER	64	0	180	0	7,920	0
14	BEALE	30,890	0	34,231	0	32,641	0
15	SHUBERT	5,766	0	4,414	0	6,439	0
16	GOLDSTEIN-PRICE	90,456,082	1.44E-09	12,742	0	10,760	0
17	SIX-HUMP CAMEL	42	0	14	0	2,003	0
18	SPHERE	831,432	0	867,536	0	5,430,227	0
19	Parallel	844,132	0	925,256	0	5,525,623	0
20	Rotated	769,933	0	90,186,647	331,995	5,503,082	0
21	CIGAR	828,087	0	947,315	0	5,379,439	0
22	BROWN	90,354,281	558.776	948,402	0	5,387,857	0
23	MULTIMOD	90,506,375	1.00E+10	1,472,982	0	14,987,233	0
24	ZAKHAROV	90,630,857	1.00E+10	90,003,965	3.06E-29	84,336,016	0
25	TRID	537,131	0	283,321	0	290,386	0
26	EASOM	15,177	0	17,886	0	18,290	0
27	EGGHOLDER	59	0	90,734,001	24.302	90,543,127	0.001781

จากตารางที่ ง.1 และ ง.2 จะเห็นว่าเมื่อมีการเปลี่ยนตำแหน่งของค่าตำแหน่งที่ดีที่สุดเป็นค่าคงที่เป็น 0.1 วิธี MSPSO มีฟังก์ชันที่ไม่พบคำตอบจำนวน 3 ฟังก์ชัน และเมื่อเปลี่ยนตำแหน่งไปที่ 0.01 และ 0.00001 มีฟังก์ชันที่ไม่พบคำตอบจำนวน 4 และ 5 ฟังก์ชันตามลำดับ ขณะที่ถ้าเปลี่ยนตำแหน่งไปทาง 25% ของ Upper Limit จะมี 7 ฟังก์ชันที่ไม่พบคำตอบ แต่ในกรณีที่มีการเปลี่ยนตำแหน่งของค่าคำตอบเป็นค่าสุ่มพบว่ามี 8 ฟังก์ชันที่ไม่พบคำตอบ แต่ในกรณีที่พบคำตอบจะเห็นว่าโดยส่วนใหญ่เจอค่าคำตอบได้ดีกว่าโดยเฉพาะในฟังก์ชัน MULTIMOD จะเห็นได้ว่าการเปลี่ยนตำแหน่งมีผลต่อวิธี MSPSO ทำให้บางฟังก์ชันค้นหาคำตอบได้เร็วขึ้น บางฟังก์ชันใช้เวลาในการค้นหานานขึ้นและบางฟังก์ชันไม่เจอคำตอบเลย

ง.2 การทดลองเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่งอื่นเพื่อเปรียบเทียบประสิทธิภาพระหว่าง GBPSO 30 อนุภาค และ 90 อนุภาค กับ MSPSO

ตาราง ง.3 ตารางแสดงผลการทดลองเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่ง 25% ของขอบเขตบนของแต่ละฟังก์ชันเพื่อเปรียบเทียบประสิทธิภาพระหว่าง GBPSO 30 อนุภาค และ 90 อนุภาค กับ MSPSO

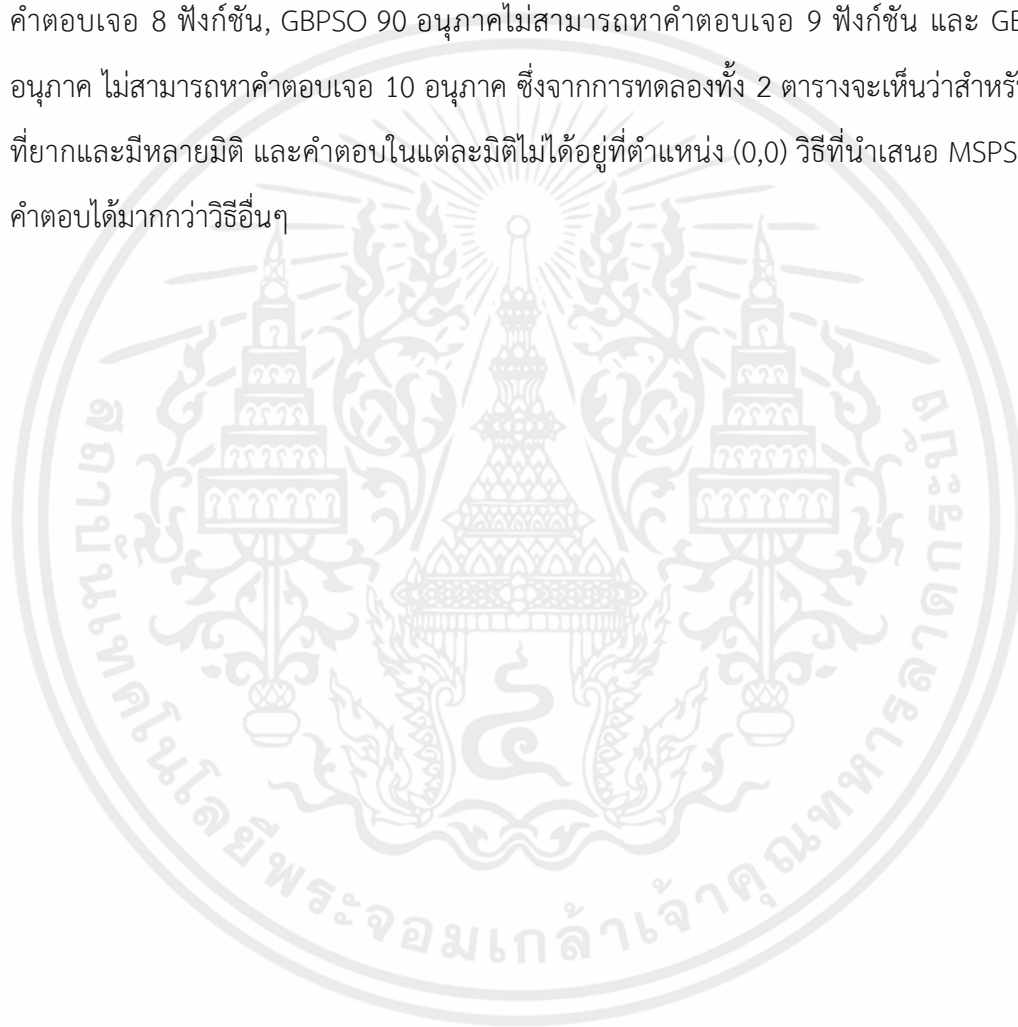
NO	Function	GBPSO (30)		GBPSO (90)		MSPSO	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	30,011,045	3.91E-14	90,122,082	2.8422E-14	90,113,228	2.84E-14
2	GRIEWANK	3,768,588	0	16,602,189	0	2,145,092	0
3	RASTRIGIN	30,015,705	1.98992	16,423,044	0	19,888,830	0
4	ROSENBROCK	30,007,521	2.47E-30	1,453,961	0	1,046,760	0
5	SCHWEFEL	773,649	0	598,345	0	253,220	0
6	COSINE MIXTURE	29,580,683	1.78E-15	3,334,295	0	4,077,945	0
7	EXPONENTIAL	30,025,260	3.33E-16	90,213,510	2.2205E-16	90,212,490	3.33E-16
8	LEVY	4,784,018	0	3,179,031	0	3,440,608	0
9	MICHALEWICZ	117,230	0	213,305	0	235,770	0
10	DIXON-PRICE	30,292,413	507.1195	90,885,557	505.169	90,892,363	505.169
11	STEP	156,122	0	170,057	0	207,654	0
12	SCHAFFER	6,646	0	314,933	0	31,611	0
13	HOLDER	27	0	71	0	64	0
14	BEALE	11,777	0	32,437	0	30,890	0
15	SHUBERT	2,166	0	6,084	0	5,766	0
16	GOLDSTEIN-PRICE	30,149,097	6.92E-08	90,415,655	1.1175E-08	90,456,082	1.44E-09
17	SIX-HUMP CAMEL	39	0	24	0	42	0
18	SPHERE	1,216,435	0	853,324	0	831,432	0
19	Parallel	561,975	0	803,967	0	844,132	0
20	Rotated	1,588,820	0	854,944	0	769,933	0
21	CIGAR	3,190,619	0	870,580	0	828,087	0
22	BROWN	30,140,085	6593.705	90,372,198	2,935.57	90,354,281	558.776
23	MULTIMOD	30,157,148	5E+09	90,460,093	1E+10	90,506,375	1.00E+10
24	ZAKHAROV	30,215,256	1E+10	90,622,881	1E+10	90,630,857	1.00E+10
25	TRID	169,527	0	196,046	0	537,131	0
26	EASOM	6,555	0	17,151	0	15,177	0
27	EGGHOLDER	83	0	58	0	59	0

ตาราง ง.4 ตารางแสดงผลการทดลองเปลี่ยนตำแหน่งของค่าคำตอบที่ดีที่สุดไปยังตำแหน่งอื่นๆแบบสุ่มโดยที่คำตอบในแต่ละมิติมีค่าที่ไม่เท่ากัน(Random Shift)เพื่อเปรียบเทียบประสิทธิภาพระหว่าง GBPSO 30 อนุภาค และ 90 อนุภาค กับ MSPSO

NO	Function	GBPSO (30)		GBPSO (90)		MSPSO	
		Evaluation Call	Best Fitness	Evaluation Call	Best Fitness	Evaluation Call	Best Fitness
1	ACKLEY	30,011,279	4.61853E-14	90,127,597	2.84217E-14	90,116,970	2.84E-14
2	GRIEWANK	20,050,143	0	<u>435,057</u>	0	5,359,618	0
3	RASTRIGIN	30,022,767	0.99496	90,190,333	0.994959	<u>15,070,312</u>	0
4	ROSENBROCK	30,007,208	1.30655E-28	90,099,176	1.54074E-29	90,087,871	1.48E-29
5	SCHWEFEL	30,023,356	4,665.21	90,069,904	2,963.49	90,063,065	4623.63
6	COSINE MIXTURE	30,027,672	1.77636E-15	2,784,075	0	<u>1,758,777</u>	0
7	EXPONENTIAL	30,030,574	4.44089E-16	90,225,885	2.22045E-16	90,222,664	3.33E-16
8	LEVY	3,891,261	0	2,852,403	0	<u>1,663,571</u>	0
9	MICHALEWICZ	162,463	0	250,286	0	<u>135,882</u>	0
10	DIXON-PRICE	30,169,389	0.511376	90,769,322	0.1066281	90,609,356	0.090251
11	STEP	142,355	0	159,208	0	<u>132,324</u>	0
12	SCHAFFER	269,188	0	19,696	0	<u>17,935</u>	0
13	HOLDER	<u>46</u>	0	183	0	180	0
14	BEALE	<u>11,201</u>	0	31,671	0	34,231	0
15	SHUBERT	<u>1,977</u>	0	6,533	0	4,414	0
16	GOLDSTEIN-PRICE	<u>3,808</u>	0	11,163	0	12,742	0
17	SIX-HUMP CAMEL	47	0	283	0	<u>14</u>	0
18	SPHERE	4,027,797	0	<u>825,767</u>	0	867,536	0
19	Parallel	1,739,006	0	<u>833,126</u>	0	925,256	0
20	Rotated	30,032,429	251,895.50	90,175,049	290,769	90,186,647	331,995
21	CIGAR	940,544	0	<u>807,307</u>	0	947,315	0
22	BROWN	9,166,348	0	<u>843,343</u>	0	948,402	0
23	MULTIMOD	4,976,291	0	1,476,241	0	<u>1,472,982</u>	0
24	ZAKHAROV	30,000,250	9.18078E-24	90,004,479	8.62E-30	90,003,965	3.06E-29
25	TRID	<u>156,199</u>	0	287,507	0	283,321	0
26	EASOM	<u>6,840</u>	0	17,854	0	17,886	0
27	EGGHOLDER	15,246,915	9.2369	90,335,863	2.72177	90,734,001	24.302

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ ง.3 และ ง.4 จะเห็นว่าเมื่อมีการเปลี่ยนตำแหน่งของค่าตำแหน่งที่ดีที่สุดไปเป็นค่าอื่นทำให้มีหลายฟังก์ชันไม่สามารถหาค่าคำตอบ(ดูเปรียบเทียบกับตารางที่ 5.3 และ 5.4 ในบทที่ 5) โดยในตาราง ง.3 เป็นการเปลี่ยนตำแหน่งไปที่ 25% ของขอบเขตบนในแต่ละฟังก์ชัน จะเห็นว่า MSPSO และ GBPSO 90 อนุภาคไม่สามารถหาค่าคำตอบ 7 ฟังก์ชัน ขณะที่ GBPSO 30 อนุภาคไม่สามารถหาค่าคำตอบ 10 อนุภาค และในตาราง ง.4 เป็นการเปลี่ยนตำแหน่งไปที่ตำแหน่งอื่น ๆ ด้วยการสุ่มโดยที่คำตอบในแต่ละมิติมีค่าที่ไม่เท่ากัน(Random Shift) จะเห็นว่า MSPSO ไม่สามารถหาค่าคำตอบ 8 ฟังก์ชัน, GBPSO 90 อนุภาคไม่สามารถหาค่าคำตอบ 9 ฟังก์ชัน และ GBPSO 30 อนุภาค ไม่สามารถหาค่าคำตอบ 10 อนุภาค ซึ่งจากการทดลองทั้ง 2 ตารางจะเห็นว่าสำหรับฟังก์ชันที่ยากและมีหลายมิติ และคำตอบในแต่ละมิติไม่ได้อยู่ที่ตำแหน่ง (0,0) วิธีที่นำเสนอ MSPSO จะพบคำตอบได้มากกว่าวิธีอื่นๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล นางสาวลัคนา เพิ่มพูล
 วัน เดือน ปีเกิด 16 กรกฎาคม ที่ฉะเชิงเทรา
 ประวัติการศึกษา 2554 ระดับอุดมศึกษา
 คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 2550 ระดับมัธยมศึกษา
 โรงเรียนเบญจมราชรังสฤษฎิ์ ฉะเชิงเทรา

ประสบการณ์การทำงาน
 พ.ศ.2555-2559 ตำแหน่งวิศวกรระบบ (System Engineer)
 บริษัท ดับเบิล เอ (1991) จำกัด (มหาชน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้