

การปรับปรุงการหาค่าเหมาะสมแบบอนุภาคกลุ่มด้วยการสุ่มรอบอนุภาค
และกระจายอนุภาค

IMPROVEMENTS OF PARTICLE SWARM OPTIMIZATION WITH SAMPLINGS
PARTICLE PROXIMITY AND PARTICLE REDISTRIBUTION

เจียบวุฒิ รัตนวิไลสกุล
CHAIBWOOT RATANAVILISAGUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2558

KMITL-2015-EN-D-018-014

การปรับปรุงการหาค่าเหมาะสมแบบอนุภาคกลุ่มด้วยการสุ่มรอบอนุภาค
และกระจายอนุภาค

IMPROVEMENTS OF PARTICLE SWARM OPTIMIZATION WITH SAMPLINGS
PARTICLE PROXIMITY AND PARTICLE REDISTRIBUTION



T139554

เจียบวุฒิ รัตนวิไลสกุล
CHAIBWOOT RATANAVILISAGUL

เลขหมู่.....
เลขทะเบียน..... **139554**
วันเดือนปี..... **10 พ.ย. 2558**

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2558

KMITL-2015-EN-D-018-014

IMPROVEMENTS OF PARTICLE SWARM OPTIMIZATION WITH SAMPLINGS
PARTICLE PROXIMITY AND PARTICLE REDISTRIBUTION

CHAIBWOOT RATANAVILISAGUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2015
KMITL-2015-EN-D-018-014

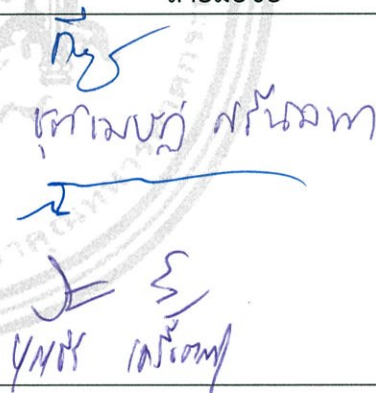
COPYRIGHT 2015

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การปรับปรุงการหาค่าเหมาะสมแบบอนุภาคกลุ่มด้วยการสุ่มรอบอนุภาคและการกระจายอนุภาค
Thesis Title Improvements of Particle Swarm Optimization with Samplings Particle Proximity and Particle Redistribution
นักศึกษ นายเฉียบวุฒิ รัตน์วิไลสกุล
รหัสประจำตัว 56601011
ปริญญา วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.บุญธีร์ เครือตราชู
หมายเลขวิทยานิพนธ์ KMITL-2015-EN-D-018-014

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.เกียรติคุณ	เจียรนัยธนะกิจ	
ผศ.ดร.ชุตินเมษมณี	ศรีนิลทา	
รศ.ดร.นิพนธ์	เจริญกิจการ	
รศ.ดร.ปิติเขต	สุรักษา	
รศ.ดร.บุญธีร์	เครือตราชู	

วัน / เดือน/ ปี ที่สอบ วันจันทร์ที่ 16 กุมภาพันธ์ พ.ศ. 2558 เวลา 09.00-11.00 น.
สถานที่สอบ ณ อาคาร A ชั้น 5 ห้องประชุม 4

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร. คมสัน มาลีสี)

คณบดี คณะวิศวกรรมศาสตร์
วันที่ 16 กุมภาพันธ์ พ.ศ. 2558

หัวข้อวิทยานิพนธ์	การปรับปรุงการหาค่าเหมาะสมแบบอนุภาคกลุ่มด้วยการสุ่มรอบอนุภาคและกระจายอนุภาค
นักศึกษา	นายเฉียบวุฒิ รัตนวิไลสกุล
รหัสนักศึกษา	56601011
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2558
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.บุญธีร์ เครือตราชู

บทคัดย่อ

ปัญหาหลักของการหาค่าเหมาะสมแบบอนุภาคกลุ่ม (Particle Swarm Optimization หรือ PSO) คือปัญหาการติดในจุดสุดสัมพัทธ์ (Trapping in Local Optimum) และปัญหาการลู่เร็วเกินไป (Premature Convergence) โดยงานวิทยานิพนธ์ฉบับนี้มีจุดประสงค์ที่จะพัฒนาวิธีการที่นำมาใช้ลดปัญหาทั้งสองซึ่งได้นำวิธีการกลายพันธุ์รอบอนุภาคร่วมกับวิธีการรีโพซิชั่น โดยการประยุกต์วิธีการกลายพันธุ์ที่นำเสนอร่วมกับวิธีการรีโพซิชั่นที่นำเสนอ ได้พิสูจน์ให้เห็นว่าสามารถลดปัญหาทั้งสองของ PSO และเพิ่มประสิทธิภาพการค้นหาคำตอบของ PSO ดังนั้นวิธีนี้จึงเป็นวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ โดยเรียกวิธีที่นำเสนอชื่อว่า การประยุกต์การกลายพันธุ์และรีโพซิชั่นกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Applying Mutation and Reposition to Particle Swarm Optimization หรือ MRPSO) สำหรับการทดลองของ MRPSO ได้ทดสอบกับปัญหาฟังก์ชันมาตรฐาน (Benchmark Functions) 24 ฟังก์ชัน ปัญหาถุงเป้แบบหลายมิติ (Multidimensional Knapsack Problems หรือ MKP) และปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem หรือ TSP) พบว่า MRPSO ได้ผลลัพธ์การค้นหาคำตอบที่ดีกว่า PSO เทคนิคอื่นทั้งในปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้แบบหลายมิติ และปัญหาการเดินทางของพนักงานขาย

Thesis Title	Improvements of particle swarm optimization with samplings particle proximity and particle redistribution
Student	Mr. Chiabwoot Ratanavilisagul
Student ID.	56601011
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2015
Thesis Advisor	Assoc. Prof. Dr. Boontee Kruatrachue

ABSTRACT

The common problems of particle swarm optimization (PSO) are those of trapping in local optimum and premature convergence. This thesis aims to develop a solution to both problems by introducing mutation around particles and employing the reposition technique. The concurrent use of the introduced mutation and reposition has proved to solve both problems and enhanced the PSO performance; and thus is employed in this dissertation. The proposed technique is termed Applying Mutation and Reposition to Particle Swarm Optimization (MRPSO). MRPSO is tested on twenty-four benchmark functions, the multidimensional knapsack problems (MKP) and the traveling salesman problem (TSP). MRPSO yields more satisfactory search results than PSOs for the benchmark functions, MKPs and TSPs.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ เนื่องจากได้รับความกรุณาเป็นอย่างสูงจาก อาจารย์ผู้ควบคุมวิทยานิพนธ์ คือ รองศาสตราจารย์ ดร. บุญธีร์ เครือตราชู ที่ช่วยให้คำปรึกษาเป็นอย่างดีเสมอมา อีกทั้งยังให้วิชาความรู้ในหัวข้อต่างๆ ที่เกี่ยวข้องกับงานวิจัยตั้งแต่เริ่มต้นจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ต้องขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

ขอขอบพระคุณกรรมการสอบวิทยานิพนธ์ทุกท่านที่ได้กรุณาให้คำแนะนำต่างๆ จนทำให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์ สามารถครอบคลุมเนื้อหาของงานวิจัยในแง่มุมต่างๆ อย่างครบถ้วน

ขอขอบพระคุณอาจารย์ทุกท่าน ที่ได้ให้ความรู้ต่างๆ ทั้งโดยตรงและโดยอ้อม ตลอดจนนักวิจัยทุกท่านที่เอื้อเฟื่องานวิจัยจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ทุนการศึกษา ระยะเวลาเล่าเรียนซึ่งเป็นการสนับสนุนในการทำวิจัยจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพที่ให้ทุนการศึกษา ทำให้ช่วยสนับสนุนในการทำวิจัยจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคนที่อยู่ในห้องวิจัยที่คอยให้ความช่วยเหลือสนับสนุนในเรื่องต่างๆ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณส่วนสนับสนุนวิชาการและบัณฑิตศึกษา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้การสนับสนุนในการทำวิจัยจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายขอขอบพระคุณมารดาของข้าพเจ้าที่ได้ให้การสนับสนุนในเรื่องต่างๆ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

หวังว่าผู้อ่านคงจะได้ความรู้จากวิทยานิพนธ์ฉบับนี้ไม่มากก็น้อย และหากวิทยานิพนธ์ฉบับนี้มีข้อผิดพลาดประการใด ข้าพเจ้าขอน้อมรับไว้แต่เพียงผู้เดียว

เฉียววุฒิ รัตนวิไลสกุล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	X
สารบัญรูป.....	XII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา	4
1.3 สมมติฐานของการศึกษา	4
1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย	5
1.5 ขอบเขตการวิจัย	5
1.6 ขั้นตอนการศึกษา.....	6
1.7 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย	6
1.8 โครงสร้างของวิทยานิพนธ์.....	7
บทที่ 2 ทฤษฎีพื้นฐานที่เกี่ยวข้อง.....	8
2.1 ปัญหาการหาจุดที่ดีที่สุด	8
2.1.1 นิยามของปัญหาการหาจุดที่ดีที่สุด.....	8
2.1.1.1 ปัญหาการหาจุดที่ดีที่สุด	8
2.1.1.2 ฟังก์ชันเป้าหมาย	8
2.1.1.3 ตัวแปรต้น.....	8
2.1.1.4 ข้อจำกัด.....	9
2.1.2 ประเภทของปัญหาการหาจุดที่ดีที่สุด.....	9
2.1.2.1 ปัญหาการหาจุดที่ดีที่สุดที่ต้องการหาค่าสูงสุดและค่าต่ำสุด	9
2.1.2.2 ปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่องและเชิงตัดสินใจ.....	9
2.1.2.3 ปัญหาการหาจุดที่ดีที่สุดฐานนิยมเดียวและหลายรูปแบบ.....	10
2.1.3 ตัวอย่างปัญหาการหาจุดที่ดีที่สุด	11
2.2 การหาจุดที่ดีที่สุด.....	12
2.2.1 นิยามของการหาจุดที่ดีที่สุด.....	12
2.2.2 ประเภทของขั้นตอนวิธีการหาจุดที่ดีที่สุด.....	12
2.2.2.1 ประเภทขั้นตอนวิธีพื้นฐานเส้นโคจรและประชากร.....	12
2.2.2.2 ประเภทขั้นตอนวิธีการค้นหาอย่างแน่นอนและไม่แน่นอน.....	12
2.2.2.3 ประเภทขั้นตอนวิธีใช้และไม่ใช้ผลการค้นหาเดิม	13

สารบัญ (ต่อ)

	หน้า
2.2.3 พฤติกรรมการค้นหาของวิธีการหาจุดที่ดีที่สุด	13
2.3 ปัญหาการติดในจุดสุดสัมพัทธ์.....	13
2.3.1 จุดสุดสัมพัทธ์และจุดสุดสัมบูรณ์.....	13
2.3.2 นิยามของปัญหาการติดในจุดสุดสัมพัทธ์.....	14
2.3.3 ตัวอย่างของปัญหาการติดในจุดสุดสัมพัทธ์	15
2.3.4 สาเหตุของปัญหาการติดในจุดสุดสัมพัทธ์.....	16
2.4 วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค.....	16
2.4.1 นิยามของ PSO.....	16
2.4.2 นิยามที่ใช้กับ PSO.....	17
2.4.2.1 อนุภาค	17
2.4.2.2 ประชากร.....	17
2.4.2.3 รอบการค้นหาคำตอบ.....	17
2.4.2.4 ความเร็วของอนุภาค.....	17
2.4.2.5 ความเร็วสูงสุด	18
2.4.2.6 ตำแหน่งของอนุภาค	18
2.4.2.7 ตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัว.....	18
2.4.2.8 ตำแหน่งที่ดีที่สุดของอนุภาคทั้งหมด	18
2.4.2.9 ฟังก์ชันคำนวณค่าความเหมาะสมหรือฟังก์ชันเป้าหมาย.....	18
2.4.3 หลักการทำงานของ PSO.....	19
2.4.3.1 กระบวนการกำหนดค่าเริ่มต้น.....	19
2.4.3.2 กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค	19
2.4.3.3 กระบวนการเปลี่ยนแปลง PBEST.....	20
2.4.3.4 กระบวนการเปลี่ยนแปลง GBEST.....	20
2.4.3.5 กระบวนการคำนวณความเร็ว	20
2.4.3.6 กระบวนการเปลี่ยนตำแหน่ง.....	21
2.5 ตัวอย่างการประยุกต์ใช้ PSO ในทางปฏิบัติ	22
2.6 ข้อดีและข้อเสียของ PSO.....	22
2.6.1 ข้อดีของ PSO.....	22
2.6.2 ข้อเสียของ PSO	24
บทที่ 3 งานวิจัยที่เกี่ยวข้อง.....	27
3.1 การประยุกต์การกลายพันธุ์กับ PSO	27
3.1.1 นิยามของการกลายพันธุ์.....	27
3.1.2 ขั้นตอนของการกลายพันธุ์.....	28
3.1.3 องค์ประกอบของการกลายพันธุ์.....	28

สารบัญ (ต่อ)

	หน้า
3.1.3.1 คำสั่งจากการกระจายตัว.....	28
3.1.3.2 สมการการกลายพันธุ์.....	31
3.1.3.3 พารามิเตอร์ที่ถูกดำเนินการการกลายพันธุ์.....	34
3.1.3.4 รอบที่จะดำเนินการกลายพันธุ์.....	35
3.1.3.5 จำนวนการกลายพันธุ์.....	35
3.1.4 ประเภทของการกลายพันธุ์.....	37
3.1.4.1 การกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ.....	37
3.1.4.2 การกลายพันธุ์ประเภทมีการเปรียบเทียบ.....	37
3.1.5 ข้อดีของการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ.....	38
3.1.6 ข้อดีของการกลายพันธุ์ประเภทมีการเปรียบเทียบ.....	40
3.1.7 ข้อเสียของการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ.....	41
3.1.7.1 ปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ.....	41
3.1.7.2 ปัญหาการกลายพันธุ์น้อยเกินไป.....	43
3.1.8 ข้อเสียของการกลายพันธุ์ประเภทมีการเปรียบเทียบ.....	44
3.1.8.1 ปัญหาการลู่เร็วเกินไป.....	44
3.1.8.2 ปัญหาการกลายพันธุ์น้อยเกินไป.....	47
3.2 การประยุกต์วิธีการเริ่มต้นการค้นหาใหม่กับ PSO.....	48
3.2.1 นิยามของการเริ่มต้นการค้นหาใหม่.....	48
3.2.2 นิยามของการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม.....	48
3.2.3 ขั้นตอนของการเริ่มต้นการค้นหาใหม่.....	49
3.2.4 องค์ประกอบของการเริ่มต้นการค้นหาใหม่.....	49
3.2.4.1 รอบที่จะรีเซ็ต.....	49
3.2.4.2 สมการรีเซ็ต.....	51
3.2.4.3 จำนวนการรีเซ็ต.....	51
3.2.4.4 ส่วนประกอบที่ถูกรีเซ็ต.....	52
3.2.5 ข้อดีของการเริ่มต้นการค้นหาใหม่.....	52
3.2.6 ข้อดีของการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม.....	54
3.2.7 ข้อเสียของวิธีการเริ่มต้นการค้นหาใหม่.....	55
3.2.8 ข้อเสียของวิธีการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม.....	56
บทที่ 4 การปรับปรุงการกลายพันธุ์และรีโพอิชัน.....	58
4.1 การเพิ่มประสิทธิภาพของ PSO ที่ประยุกต์ใช้ CM โดยลดปัญหาการลู่เร็วเกินไป.....	58
4.2 การเพิ่มของประสิทธิภาพของ PSO หลังจากเกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	66
4.2.1 การติดในจุดสุดสัมพัทธ์ของ PSO ที่ประยุกต์ใช้ CM.....	66
4.2.2 การปรับปรุงวิธีการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม.....	67

สารบัญ (ต่อ)

	หน้า
4.3 วิธีการรีโพิซชันสำหรับ PSO ที่ประยุกต์ใช้การกลายพันธุ์.....	71
4.4 วิเคราะห์ข้อดีและข้อเสียของวิธีที่นำเสนอ.....	74
4.4.1 ข้อดีของวิธีที่นำเสนอ.....	74
4.4.2 ข้อเสียของวิธีที่นำเสนอ.....	75
บทที่ 5 การทดลองการประยุกต์การกลายพันธุ์และรีโพิซชัน.....	77
5.1 การทดลองปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO.....	84
5.2 การทดลองการประยุกต์ใช้การกลายพันธุ์กับ PSO.....	86
5.2.1 การพิจารณาการกลายพันธุ์แบบ NCM กับ PSO.....	86
5.2.2 การพิจารณาการเพิ่มประสิทธิภาพการค้นหาของ PSO ด้วย NCM.....	89
5.2.3 การพิจารณาการกลายพันธุ์แบบ CM.....	89
5.3 การทดลองปัญหาจากการกลายพันธุ์แบบ NCM.....	91
5.3.1 การพิจารณาปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ.....	91
5.3.2 การพิจารณาปัญหาการกลายพันธุ์น้อยเกินไปของ NCM.....	93
5.4 การทดลองปัญหาและแนวทางการแก้ปัญหาการกลายพันธุ์แบบ CM.....	94
5.4.1 การพิจารณาความหลากหลายของประชากรของการกลายพันธุ์แบบ CM.....	94
5.4.2 การพิจารณาปัญหาการลู่เร็วเกินไปและแนวทางแก้ปัญหาการลู่เร็วเกินไป.....	95
5.4.3 การพิจารณาปัญหาการกลายพันธุ์น้อยเกินไปของ CM.....	98
5.5 การทดลองการประยุกต์ใช้รีรันและรีโพิซชัน.....	100
5.6 การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพิซชัน.....	103
5.7 การทดลองผลกระทบจากการกลายพันธุ์และรีโพิซชัน.....	105
5.8 การทดลองขั้นตอนวิธีที่นำเสนอกับฟังก์ชันมาตรฐาน.....	107
5.9 การทดลองขั้นตอนวิธีที่นำเสนอกับปัญหาถุงเป้หลายมิติ.....	108
5.10 การทดลองขั้นตอนวิธีที่นำเสนอกับปัญหาการเดินทางของพนักงานขาย.....	110
บทที่ 6 สรุปผลการทดลองและข้อเสนอแนะ.....	112
6.1 สรุปผลการทดลอง.....	112
6.2 ข้อเสนอแนะ.....	114
เอกสารอ้างอิง.....	116
ภาคผนวก.....	125
ภาคผนวก ก. รายละเอียดของฟังก์ชันมาตรฐาน.....	126

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ข. การประยุกต์ PSO กับปัญหาถุงเป้แบบหลายมิติ.....	143
ข.1 ปัญหาถุงเป้แบบหลายมิติ.....	143
ข.2 การประยุกต์ BPSO กับปัญหาถุงเป้แบบหลายมิติ	144
ข.2.1 กระบวนการกำหนดค่าเริ่มต้น	144
ข.2.2 กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค	145
ข.2.3 กระบวนการเปลี่ยนแปลง PBEST.....	145
ข.2.4 กระบวนการเปลี่ยนแปลง GBEST	145
ข.2.5 กระบวนการคำนวณความเร็ว.....	146
ข.2.6 กระบวนการเปลี่ยนตำแหน่ง.....	146
ข.3 การประยุกต์ NCM และรีโพอิชันกับ BPSO กับปัญหาถุงเป้แบบหลายมิติ	147
ข.4 การประยุกต์ CM กับ BPSO กับปัญหาถุงเป้แบบหลายมิติ	148
ภาคผนวก ค. การประยุกต์ PSO กับปัญหาการเดินทางของพนักงานขาย.....	149
ค.1 รูปแบบคำตอบของปัญหาการเดินทางของพนักงานขาย	149
ค.2 ขั้นตอนการสลับที่คำตอบของปัญหาการเดินทางของพนักงานขาย	150
ค.3 การประยุกต์ PPSO กับปัญหาการเดินทางของพนักงานขาย	151
ค.3.1 กระบวนการกำหนดค่าเริ่มต้น	151
ค.3.2 กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค	152
ค.3.3 กระบวนการเปลี่ยนแปลง PBEST.....	152
ค.3.4 กระบวนการเปลี่ยนแปลง GBEST	153
ค.3.5 กระบวนการคำนวณความเร็ว.....	153
ค.3.6 กระบวนการเปลี่ยนตำแหน่ง.....	154
ค.4 การประยุกต์ NCM และรีโพอิชันกับ PPSO กับปัญหาการเดินทางของพนักงานขาย	154
ค.5 การประยุกต์ CM กับ PPSO กับปัญหาการเดินทางของพนักงานขาย	155
ภาคผนวก ง. การทดลองการประยุกต์การกลายพันธุ์และรีโพอิชันกับ MKP.....	157
ง.1 การทดลองปัญหาการตีในจุดสุดสัมพัทธ์	159
ง.2 การทดลองการประยุกต์ใช้การกลายพันธุ์.....	161
ง.2.1 การพิจารณาการกลายพันธุ์แบบ NCM.....	161
ง.2.2 การพิจารณาการเพิ่มประสิทธิภาพการค้นหาของ PSO ด้วย NCM	164
ง.2.3 การพิจารณาการกลายพันธุ์แบบ CM	165
ง.3 การทดลองปัญหาจากการกลายพันธุ์แบบ NCM.....	167
ง.3.1 การพิจารณาปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ	168
ง.4 การทดลองปัญหาและแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM.....	169
ง.4.1 การพิจารณาความหลากหลายของประชากรของการกลายพันธุ์แบบ CM	169

สารบัญ (ต่อ)

	หน้า
ง.4.2 การพิจารณาปัญหาการลู่เร็วเกินไปและแนวทางแก้ปัญหาการลู่เร็วเกินไป	170
ง.5 การทดลองการประยุกต์ใช้รีรันและรีโพซิชั่น.....	174
ง.6 การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพซิชั่น.....	177
ภาคผนวก จ. การทดลองการประยุกต์การกลายพันธุ์และรีโพซิชั่นกับ TSP	180
จ.1 การทดลองปัญหาการติดในจุดสุดสัมพัทธ์	181
จ.2 การทดลองการประยุกต์ใช้การกลายพันธุ์.....	182
จ.2.1 การพิจารณาการกลายพันธุ์แบบ NCM	183
จ.2.2 การพิจารณาการเพิ่มประสิทธิภาพการค้นหาของ PSO ด้วย NCM.....	184
จ.2.3 การพิจารณาการกลายพันธุ์แบบ CM.....	185
จ.3 การทดลองปัญหาจากการกลายพันธุ์แบบ NCM	187
จ.3.1 การพิจารณาปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ.....	187
จ.4 การทดลองปัญหาและแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM.....	187
จ.4.1 การพิจารณาความหลากหลายของประชากรของการกลายพันธุ์แบบ CM.....	187
จ.4.2 การพิจารณาปัญหาการลู่เร็วเกินไปและแนวทางแก้ปัญหาการลู่เร็วเกินไป.....	188
จ.5 การทดลองการประยุกต์ใช้รีรันและรีโพซิชั่น	191
จ.6 การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพซิชั่น	193
ภาคผนวก ฉ. รหัสจำลองของขั้นตอนวิธีการกลายพันธุ์และรีโพซิชั่น	195
ภาคผนวก ช. งานวิจัยที่ได้รับการตีพิมพ์	215
ภาคผนวก ซ. การทดลองพารามิเตอร์ของการกลายพันธุ์และรีโพซิชั่น	253
ซ.1 การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG กับฟังก์ชันมาตรฐาน	253
ซ.2 การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG กับ MKP	257
ซ.3 การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG กับ TSP	263
ภาคผนวก ฅ. วิธีการกลายพันธุ์และรีโพซิชั่นแบบอื่นที่นำมาเปรียบเทียบกับวิธีการที่นำเสนอ	266
ฅ.1 วิธีการกลายพันธุ์แบบอื่นที่นำมาเปรียบเทียบกับวิธีการกลายพันธุ์ที่นำเสนอ.....	266
ฅ.2 การติดในจุดสุดสัมพัทธ์ของ PSO ที่ประยุกต์ใช้ CM	271
ฅ.3 รีโพซิชั่นแบบอื่นที่นำมาเปรียบเทียบกับรีโพซิชั่นที่นำเสนอ	273
ฅ.4 ปัญหาของรีโพซิชั่น	275
ประวัติผู้เขียน	278

สารบัญตาราง

ตารางที่	หน้า
5.1 พารามิเตอร์ที่ใช้ทดลองกับฟังก์ชันคณิตศาสตร์.....	80
5.2 พารามิเตอร์ที่ใช้ทดลองกับปัญหาถุงเป้หลายมิติ.....	81
5.3 พารามิเตอร์ที่ใช้ทดลองกับปัญหาการเดินทางของพนักงานขาย.....	81
5.4 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO.....	85
5.5 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO และ FIM.....	88
5.6 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO และ MXUX.....	91
5.7 ผลการทดลองปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ.....	92
5.8 ผลการทดลองปัญหาการกลายพันธุ์น้อยเกินไปสำหรับ NCM.....	94
5.9 ผลการทดลองปัญหาการลู่เร็วเกินไป.....	97
5.10 ผลการทดลองปัญหาการกลายพันธุ์น้อยเกินไปสำหรับ CM.....	100
5.11 ผลของ ARR ของฟังก์ชันมาตรฐานด้วย RERUN และ REPOSITION.....	102
5.12 ผลการค้นหาคำตอบของฟังก์ชันคณิตศาสตร์ด้วย PSO, RERUN และ REPOSITION.....	102
5.13 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO, RG, RP, RN, RPG.....	105
5.14 ผลการทดลองฟังก์ชันมาตรฐาน.....	107
5.15 ผลการทดลองปัญหาถุงเป้หลายมิติ.....	109
5.16 ผลการทดลองปัญหาการเดินทางของพนักงานขาย.....	111
ก.1 สรุปสมบัติของฟังก์ชันคณิตศาสตร์ที่ใช้ในการทดลอง.....	127
ง.1 สรุปสมบัติของปัญหาถุงเป้แบบหลายมิติที่ใช้ในการทดลอง.....	158
ง.2 ผลการค้นหาคำตอบของปัญหาถุงเป้แบบหลายมิติด้วย BPSO.....	160
ง.3 ผลการค้นหาคำตอบของปัญหาถุงเป้หลายมิติด้วย BPSO และ FIM.....	163
ง.4 ผลการค้นหาคำตอบของปัญหาถุงเป้หลายมิติด้วย BPSO และ MXUX.....	167
ง.5 ผลการทดลองปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบสำหรับ MKP.....	168
ง.6 ผลการทดลองปัญหาการลู่เร็วเกินไปสำหรับ MKP.....	172
ง.7 ผลการค้นหาคำตอบด้วย BPSO, MGUG, MXUX, MPUP และ MXUPG กับ MKP.....	173
ง.8 ผลของ ARR ของปัญหาถุงเป้หลายมิติด้วย RERUN และ REPOSITION.....	175
ง.9 ผลการค้นหาคำตอบของปัญหาถุงเป้หลายมิติด้วย BPSO, RERUN และ REPOSITION.....	176
ง.10 ผลการค้นหาคำตอบของปัญหาถุงเป้หลายมิติด้วย BPSO, RG, RP, RN, RPG.....	179
จ.1 สรุปสมบัติของปัญหาการเดินทางของพนักงานขายที่ใช้ในการทดลอง.....	181
จ.2 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO.....	181
จ.3 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO และ FIM.....	184
จ.4 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO และ MXUX.....	186
จ.5 ผลการทดลองปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบสำหรับ TSP.....	187
จ.6 ผลการทดลองปัญหาการลู่เร็วเกินไปสำหรับ TSP.....	190
จ.7 ผลการค้นหาคำตอบของ TSP ด้วย PPSO, MGUG, MXUX, MPUP และ MXUPG.....	190
จ.8 ผลของ ARR ของปัญหาการเดินทางของพนักงานขายด้วย RERUN และ REPOSITION.....	192

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
จ.9 ผลการค้นหาคำตอบของ TSP ด้วย PPSO, RERUN และ REPOSITION.....	192
จ.10 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO, RG, RP, RN, RPG.....	194
ช.1 ผลการค้นหาคำตอบการปรับค่า RM กับฟังก์ชันมาตรฐานด้วย MXUPG	254
ช.2 ผลการค้นหาคำตอบการปรับค่า PM กับฟังก์ชันมาตรฐานด้วย MXUPG	255
ช.3 ผลการค้นหาคำตอบการปรับค่า TR กับฟังก์ชันมาตรฐานด้วย RPG.....	256
ช.4 ผลการค้นหาคำตอบการปรับค่า PR กับฟังก์ชันมาตรฐานด้วย RPG.....	257
ช.5 ผลการค้นหาคำตอบการปรับค่า RM กับ MKP ด้วย MXUPG.....	258
ช.6 ผลการค้นหาคำตอบการปรับค่า PM กับ MKP ด้วย MXUPG.....	259
ช.7 ผลการค้นหาคำตอบการปรับค่า TR กับ MKP ด้วย RPG	260
ช.8 ผลการค้นหาคำตอบการปรับค่า PR กับ MKP ด้วย RPG	261
ช.9 ผลการค้นหาคำตอบการปรับค่า RM กับ TSP ด้วย MXUPG.....	263
ช.10 ผลการค้นหาคำตอบการปรับค่า PM กับ TSP ด้วย MXUPG.....	263
ช.11 ผลการค้นหาคำตอบการปรับค่า TR กับ TSP ด้วย RPG.....	264
ช.12 ผลการค้นหาคำตอบการปรับค่า PR กับ TSP ด้วย RPG.....	264

สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างปัญหาถุงเป้.....	10
2.2 ตัวอย่างปัญหาการเดินทางของพนักงานขาย.....	10
2.3 ตัวอย่างปัญหายูนิโมคูล.....	11
2.4 ตัวอย่างปัญหามัลติโมคูล.....	11
2.5 จุดสุดสัมพัทธ์และจุดสุดสัมบูรณ์.....	14
2.6 ตัวอย่างของปัญหามัลติโมคูลในมุมมองสามมิติ.....	15
2.7 ตัวอย่างของปัญหามัลติโมคูลในมุมมองสองมิติ.....	15
2.8 ตัวอย่างกรณีศึกษาปัญหาการติดในจุดสุดสัมพัทธ์.....	15
2.9 ตำแหน่งอนุภาคในแกนสามมิติ.....	18
2.10 แผนภาพการทำงานของ PSO.....	19
2.11 ตัวอย่างทิศทางความเร็วของอนุภาคในแกนสองมิติ.....	21
2.12 ตัวอย่างการนำ PSO ประยุกต์ใช้ในทางปฏิบัติ.....	22
2.13 พฤติกรรมของการเคลื่อนที่ของ PSO.....	24
2.14 ความเร็วของ PSO กับปัญหาการติดในจุดสุดสัมพัทธ์.....	25
2.15 ความหลากหลายของประชากรของ PSO กับปัญหาการติดในจุดสุดสัมพัทธ์.....	25
2.16 GBEST ของ PSO กับปัญหาการติดในจุดสุดสัมพัทธ์.....	26
3.1 แผนภาพขั้นตอนการประยุกต์การกลายพันธุ์กับ PSO.....	28
3.2 การกระจายเอกรูป.....	29
3.3 การกระจายตัวปรกติ.....	29
3.4 การกระจายตัวโคซี.....	30
3.5 การกระจายยกกำลัง.....	31
3.6 ความหลากหลายของ PSO และ NCM.....	38
3.7 ความเร็วของอนุภาคของ NCM.....	38
3.8 ความเร็วของอนุภาคของ PSO.....	39
3.9 ตัวอย่าง NCM ช่วยปรับปรุง PSO.....	39
3.10 ตัวอย่าง NCM ช่วยปรับปรุง PSO.....	40
3.11 ความหลากหลายของ PSO และ CM.....	40
3.12 ตัวอย่าง CM ช่วยปรับปรุง PSO.....	41
3.13 ตัวอย่างปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ.....	42
3.14 กราฟแสดงตัวอย่างปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ.....	42
3.15 ตัวอย่างปัญหาการกลายพันธุ์น้อยเกินไปของ NCM.....	43
3.16 ตัวอย่างปัญหาการกลายพันธุ์น้อยเกินไปของ NCM.....	44
3.17 ตัวอย่างปัญหาการลู่เร็วเกินไปของ CM.....	45
3.18 การลู่เข้าหากันของ PSO.....	46
3.19 การลู่เข้าหากันของ CM.....	46

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20	ประสิทธิภาพการทำงานของขั้นตอนวิธีวิวัฒนาการ.....47
3.21	ตัวอย่างปัญหาการกระจายพันธุ์น้อยเกินไปของ.....47
3.22	แผนภาพขั้นตอนการทำงานของรีรันหรือรีโพอิชัน.....49
3.23	ตัวอย่าง RERUN ช่วยปรับปรุงคำตอบของ PSO.....53
3.24	ตัวอย่าง RERUN.....53
3.25	ตัวอย่าง Reposition ช่วยปรับปรุงคำตอบของ PSO.....54
3.26	ตัวอย่าง Reposition.....55
4.1	สถิติขอบเขตของ PSO และขอบเขตของ CM.....59
4.2	สถิติการลู่เข้าหากันของ PSO.....59
4.3	สถิติปัญหาการลู่เร็วเกินไปของ CM.....60
4.4	วิธีการกระจายพันธุ์ของ MXUX.....61
4.5	วิธีการปรับปรุงตำแหน่งอนุภาคของ MXUX.....61
4.6	การลู่เข้าหากันของ PSO.....62
4.7	การลู่เข้าหากันของ MXUX.....62
4.8	ความหลากหลายของประชากรของ PSO, MGUG, MPUP, MXUX และ MXUPG.....63
4.9	สถิติแนวทางการแก้ปัญหาการลู่เร็วเกินไปของ CM.....64
4.10	วิธีการกระจายพันธุ์ของ MXUPG และการปรับปรุง PBEST.....65
4.11	วิธีการกระจายพันธุ์ของ MXUPG และการปรับปรุง PBEST และ GBEST.....65
4.12	การลู่เข้าหากันของ MXUPG.....65
4.13	สถิติปัญหาขอบเขตการกระจายพันธุ์.....67
4.14	ตัวอย่างของ PSO.....68
4.15	ตัวอย่างของ RN.....68
4.16	ตัวอย่างของ RPG.....68
4.17	สถิติของ RN.....69
4.18	สถิติของ RPG.....70
4.19	สถิติการแก้ปัญหาขอบเขตการกระจายพันธุ์.....72
4.20	รหัสจำลองของ MRPSO.....73
4.21	ขั้นตอนการทำงานของ MRPSO.....74
5.1	LMFG ของฟังก์ชัน ACKLEY ที่ค้นหาคำตอบด้วย PSO.....85
5.2	LMFG ของฟังก์ชัน SPHERE ที่ค้นหาคำตอบด้วย PSO.....85
5.3	AVGV ของฟังก์ชัน ACKLEY ที่ค้นหาคำตอบด้วย PSO.....85
5.4	AVGV ของฟังก์ชัน SPHERE ที่ค้นหาคำตอบด้วย PSO.....85
5.5	LD(t) ของฟังก์ชัน ACKLEY ที่ค้นหาคำตอบด้วย PSO.....86
5.6	LD(t) ของฟังก์ชัน SPHERE ที่ค้นหาคำตอบด้วย PSO.....86
5.7	LMFG ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย PSO และ FIM.....87

สารบัญญรูป (ต่อ)

รูปที่	หน้า
5.8 AVGV ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย PSO	87
5.9 AVGV ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย FIM	87
5.10 D(t) ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย PSO และ FIM.....	88
5.11 FIM ปรับปรุง PSO ในช่วงรอบการค้นหา 200 ถึง 300.....	89
5.12 LMFG ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย PSO และ MXUX	90
5.13 AVGV ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย MXUX	90
5.14 D(t) ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย PSO และ MXUX.....	90
5.15 LMFx ของฟังก์ชัน ROSENBROCK ที่ค้นหาคำตอบด้วย PSO.....	92
5.16 LMFx ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย FIM และ PSO	93
5.17 ความหลากหลายของประชากรของ PSO และ CM ของฟังก์ชัน LEVY	95
5.18 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย PSO.....	95
5.19 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MGUG.....	96
5.20 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MPUP	96
5.21 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MXUX.....	96
5.22 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MXUPG.....	96
5.23 ตัวอย่างปัญหาหลายพันรุ่นน้อยเกินไปของ MGUG	98
5.24 ตัวอย่างปัญหาหลายพันรุ่นน้อยเกินไปของ MPUP.....	98
5.25 ตัวอย่างปัญหาหลายพันรุ่นน้อยเกินไปของ MXUX	99
5.26 ตัวอย่างปัญหาหลายพันรุ่นน้อยเกินไปของ MXUPG.....	99
5.27 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RERUN.....	100
5.28 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RN.....	101
5.29 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย PSO	103
5.30 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RP	103
5.31 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RG.....	104
5.32 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RPG.....	104
5.33 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย MXUPG และ RPG.....	106
5.34 LBP ของฟังก์ชัน ROSENBROCK ที่ค้นหาคำตอบด้วย MXUPG และ RPG.....	106
ก.1 ฟังก์ชัน ACKLEY.....	128
ก.2 ฟังก์ชัน GRIEWANK.....	128
ก.3 ฟังก์ชัน RASTRIGIN	129
ก.4 ฟังก์ชัน ROSENBROCK.....	129
ก.5 ฟังก์ชัน SCHWEFEL	130
ก.6 ฟังก์ชัน COSINE MIXTURE.....	131
ก.7 ฟังก์ชัน EXPONENTIAL	131
ก.8 ฟังก์ชัน LEVY.....	132

สารบัญญรูป (ต่อ)

รูปที่	หน้า
ก.9 ฟังก์ชัน MICHALEWICZ	133
ก.10 ฟังก์ชัน STEP	133
ก.11 ฟังก์ชัน SCHAFFER.....	134
ก.12 ฟังก์ชัน HOLDER.....	135
ก.13 ฟังก์ชัน BEALE	135
ก.14 ฟังก์ชัน SHUBERT	136
ก.15 ฟังก์ชัน GOLDSTEIN-PRICE.....	136
ก.16 ฟังก์ชัน TRID	137
ก.17 ฟังก์ชัน SPHERE	138
ก.18 ฟังก์ชัน PARALLEL ELLIPSOID	138
ก.19 ฟังก์ชัน ROTATED ELLIPSOID	139
ก.20 ฟังก์ชัน CIGAR.....	140
ก.21 ฟังก์ชัน BROWN.....	140
ก.22 ฟังก์ชัน MULTIMOD	141
ก.23 ฟังก์ชัน ZAKHAROV.....	141
ก.24 ฟังก์ชัน EASOM.....	142
ข.1 ตัวอย่างปัญหาถุงเป้แบบหลายมิติ	144
ข.2 ตัวอย่างการกำหนดค่าเริ่มต้นกับประชากร.....	145
ข.3 ตัวอย่างการทำงานของกระบวนการกลายพันธุ์ประเภท NCM.....	147
ข.4 ตัวอย่างการทำงานของกระบวนการกลายพันธุ์ประเภท CM	148
ค.1 ตัวอย่างรูปแบบคำตอบของ TSP.....	149
ค.2 ตัวอย่างการดำเนินการสลับที่คำตอบของ TSP	150
ค.3 ตัวอย่างการดำเนินการสลับกันของคำตอบของ TSP.....	151
ค.4 ตัวอย่างอนุภาคของ PPSO ที่นำมาประยุกต์ใช้กับ TSP.....	152
ค.5 ตัวอย่างกระบวนการคำนวณความเร็วของ PPSO	153
ค.6 รหัสจำลองของการกลายพันธุ์แบบสลับโนด	154
ค.7 ตัวอย่างการทำงานกลายพันธุ์ประเภท NCM กับ TSP.....	155
ค.8 ตัวอย่างการทำงานกลายพันธุ์ประเภท CM กับ TSP	155
ง.1 ค่าความเหมาะสมของ BPSO ที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	161
ง.2 ค่าความเหมาะสมของ BPSO ที่ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	161
ง.3 ความเร็วของ BPSO ที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	161
ง.4 ความเร็วของ BPSO ที่ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	161
ง.5 LMFG ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย BPSO	162
ง.6 LMFG ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย FIM.....	162

สารบัญรูป (ต่อ)

รูปที่	หน้า
ง.7 ค่าลอการิทึมของ AVGV ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย BPSO	162
ง.8 ค่าลอการิทึมของ AVGV ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย FIM	162
ง.9 ตัวอย่างการคำนวณความหลากหลายของประชากรของ BPSO	162
ง.10 LD(t) ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย PSO และ FIM.....	164
ง.11 FIM ปรับปรุง BPSO ในช่วงรอบการค้นหา 50 ถึง 100.....	164
ง.12 LMFG ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย MXUX.....	165
ง.13 ค่าลอการิทึม AVGV ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย MXUX.....	165
ง.14 ตัวอย่างการคำนวณความหลากหลายของประชากรของ MXUX.....	166
ง.15 LD(t) ของปัญหา WEING7 ด้วย PSO และ MXUX	166
ง.16 LD(t) ของ BPSO, MGUG, MPUP, MXUX และ MXUPG ของปัญหา WEING8.....	169
ง.17 LMFx, LMFP และ LMFG ของ BPSO ของปัญหา WEING7.....	170
ง.18 LMFx, LMFP และ LMFG ของ MGUG ของปัญหา WEING7.....	170
ง.19 LMFx, LMFP และ LMFG ของ MPUP ของปัญหา WEING7.....	170
ง.20 LMFx, LMFP และ LMFG ของ MXUX ของปัญหา WEING7.....	171
ง.21 LMFx, LMFP และ LMFG ของ MXUPG ของปัญหา WEING7.....	171
ง.22 LBP ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย RERUN	174
ง.23 LBP ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย RN.....	174
ง.24 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย BPSO	178
ง.25 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย RP	178
ง.26 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย RG.....	178
ง.27 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย RPG.....	178
จ.1 ค่าความเหมาะสมของ PPSO ที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	182
จ.2 ค่าความเหมาะสมของ PPSO ที่ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์.....	182
จ.3 LMFG ของปัญหา BAYS29 ที่ค้นหาคำตอบด้วย PPSO	183
จ.4 LMFG ของปัญหา BAYS29 ที่ค้นหาคำตอบด้วย FIM	183
จ.5 LD(t) ปัญหา BAYS29 ที่ค้นหาคำตอบด้วย PPSO และ FIM	183
จ.6 ตัวอย่างการคำนวณความหลากหลายของประชากรของ PPSO	184
จ.7 FIM ปรับปรุง PPSO ในช่วงรอบการค้นหา 2400 ถึง 2700.....	185
จ.8 ตัวอย่างการคำนวณความหลากหลายของประชากรของ MXUX.....	185
จ.9 LMFG ของปัญหา BAY29 ด้วย MXUX	186
จ.10 LD(t) ของปัญหา BAY29 ด้วย PPSO และ MXUX.....	186
จ.11 LD(t) ของ PPSO, MGUG, MPUP, MXUX และ MXUPG ของปัญหา BAYS29.....	188
จ.12 LMFx, LMFP และ LMFG ของ PPSO ของปัญหา BAYG29.....	188
จ.13 LMFx, LMFP และ LMFG ของ MGUG ของปัญหา BAYG29	189
จ.14 LMFx, LMFP และ LMFG ของ MPUP ของปัญหา BAYG29.....	189

สารบัญรูป (ต่อ)

รูปที่	หน้า
จ.15 LMFx, LMFP และ LMFG ของ MXUX ของปัญหา BAYG29	189
จ.16 LMFx, LMFP และ LMFG ของ MXUPG ของปัญหา BAYG29	189
จ.17 LBP ของ RERUN กับปัญหา BAYS29	191
จ.18 LBP ของ RN กับปัญหา BAYS29.....	191
จ.19 LBP ของ PPSO กับปัญหา BAYS29.....	193
จ.20 LBP ของ RP กับปัญหา BAYS29.....	193
จ.21 LBP ของ RG กับปัญหา BAYS29.....	193
จ.22 LBP ของ RPG กับปัญหา BAYS29	193
ฉ.1 รหัสจำลองของขั้นตอนวิธี FIM สำหรับปัญหาฟังก์ชันมาตรฐาน	195
ฉ.2 รหัสจำลองของขั้นตอนวิธี MGUG สำหรับปัญหาฟังก์ชันมาตรฐาน	195
ฉ.3 รหัสจำลองของขั้นตอนวิธี MPUP สำหรับปัญหาฟังก์ชันมาตรฐาน.....	196
ฉ.4 รหัสจำลองของขั้นตอนวิธี MXUX สำหรับปัญหาฟังก์ชันมาตรฐาน	197
ฉ.5 รหัสจำลองของขั้นตอนวิธี MXUPG สำหรับปัญหาฟังก์ชันมาตรฐาน.....	198
ฉ.6 รหัสจำลองของขั้นตอนวิธี RERUN สำหรับปัญหาฟังก์ชันมาตรฐาน	198
ฉ.7 รหัสจำลองของขั้นตอนวิธี RN สำหรับปัญหาฟังก์ชันมาตรฐาน	199
ฉ.8 รหัสจำลองของขั้นตอนวิธี RG สำหรับปัญหาฟังก์ชันมาตรฐาน.....	199
ฉ.9 รหัสจำลองของขั้นตอนวิธี RP สำหรับปัญหาฟังก์ชันมาตรฐาน.....	200
ฉ.10 รหัสจำลองของขั้นตอนวิธี RPG สำหรับปัญหาฟังก์ชันมาตรฐาน	200
ฉ.11 รหัสจำลองของขั้นตอนวิธี FIM สำหรับปัญหาถุงเป้หลายมิติ	201
ฉ.12 รหัสจำลองของขั้นตอนวิธี MGUG สำหรับปัญหาถุงเป้หลายมิติ	201
ฉ.13 รหัสจำลองของขั้นตอนวิธี MPUP สำหรับปัญหาถุงเป้หลายมิติ.....	202
ฉ.14 รหัสจำลองของขั้นตอนวิธี MXUX สำหรับปัญหาถุงเป้หลายมิติ	203
ฉ.15 รหัสจำลองของขั้นตอนวิธี MXUPG สำหรับปัญหาถุงเป้หลายมิติ.....	204
ฉ.16 รหัสจำลองของขั้นตอนวิธี RERUN สำหรับปัญหาถุงเป้หลายมิติ	204
ฉ.17 รหัสจำลองของขั้นตอนวิธี RN สำหรับปัญหาถุงเป้หลายมิติ.....	205
ฉ.18 รหัสจำลองของขั้นตอนวิธี RG สำหรับปัญหาถุงเป้หลายมิติ.....	205
ฉ.19 รหัสจำลองของขั้นตอนวิธี RP สำหรับปัญหาถุงเป้หลายมิติ	206
ฉ.20 รหัสจำลองของขั้นตอนวิธี RPG สำหรับปัญหาถุงเป้หลายมิติ	206
ฉ.21 รหัสจำลองของขั้นตอนวิธี MRPSO สำหรับปัญหาถุงเป้หลายมิติ.....	207
ฉ.22 รหัสจำลองของขั้นตอนวิธี FIM สำหรับปัญหาการเดินทางของพนักงานขาย.....	208
ฉ.23 รหัสจำลองของขั้นตอนวิธี MGUG สำหรับปัญหาการเดินทางของพนักงานขาย.....	208
ฉ.24 รหัสจำลองของขั้นตอนวิธี MPUP สำหรับปัญหาการเดินทางของพนักงานขาย	209
ฉ.25 รหัสจำลองของขั้นตอนวิธี MXUX สำหรับปัญหาการเดินทางของพนักงานขาย.....	210
ฉ.26 รหัสจำลองของขั้นตอนวิธี MXUPG สำหรับปัญหาการเดินทางของพนักงานขาย	211
ฉ.27 รหัสจำลองของขั้นตอนวิธี RERUN สำหรับปัญหาการเดินทางของพนักงานขาย.....	211

สารบัญญรูป (ต่อ)

รูปที่	หน้า
ฉ.28 รหัสจำลองของขั้นตอนวิธี RN สำหรับปัญหาการเดินทางของพนักงานขาย	212
ฉ.29 รหัสจำลองของขั้นตอนวิธี RG สำหรับปัญหาการเดินทางของพนักงานขาย	212
ฉ.30 รหัสจำลองของขั้นตอนวิธี RP สำหรับปัญหาการเดินทางของพนักงานขาย	213
ฉ.31 รหัสจำลองของขั้นตอนวิธี RPG สำหรับปัญหาการเดินทางของพนักงานขาย.....	213
ฉ.32 รหัสจำลองของขั้นตอนวิธี MRPSO สำหรับปัญหาการเดินทางของพนักงานขาย.....	214
ฅ.1 วิธีการกลายพันธุ์ของ MGUG.....	266
ฅ.2 การลู่เข้าหากันของ MGUG.....	267
ฅ.3 วิธีการปรับปรุงตำแหน่งอนุภาคของ MGUG.....	267
ฅ.4 การลู่เข้าหากันของ PSO.....	268
ฅ.5 วิธีการกลายพันธุ์ของ MPUP.....	269
ฅ.6 การลู่เข้าหากันของ MPUP.....	269
ฅ.7 วิธีการปรับปรุงตำแหน่งอนุภาคของ MPUP.....	270
ฅ.8 สาธิตกำหนดขอบเขตการกลายพันธุ์มากเกินไป.....	271
ฅ.9 ฟังก์ชัน SCHWEFEL	272
ฅ.10 ผลการค้นหาค้นหาของ MXUPG และ RPG ของ SCHWEFEL	272
ฅ.11 ฟังก์ชัน ROSENBROCK.....	272
ฅ.12 ผลการค้นหาค้นหาของ MXUPG และ RPG ของ ROSENBROCK.....	272
ฅ.13 ตัวอย่างของ RP	273
ฅ.14 ตัวอย่างของ RG	273
ฅ.15 สาธิตของ RP.....	274
ฅ.16 สาธิตของ RG	274
ฅ.17 สาธิตกรณีประชากรมีความหลากหลายเมื่อติดจุดสุดสัมพัทธ์.....	276
ฅ.18 สาธิตของปัญหาขาดความหลากหลาย.....	276

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO) ถูกคิดค้นขึ้นมาครั้งแรกโดย เคนเนดี และ อีเบอร์ฮาท ในปี 1995 [1, 2] โดยพิจารณาจากพฤติกรรมการเคลื่อนไหวของฝูงสัตว์ เช่น ฝูงนก ฝูงปลา และการสื่อสารแลกเปลี่ยนข้อมูลระหว่างสัตว์ภายในฝูงเหล่านั้น โดย PSO เป็นขั้นตอนวิธี (Algorithm) ที่ใช้แก้ปัญหาการหาจุดที่ดีที่สุด (Optimization Problems) ที่สามารถแทนด้วยจุดบนปริภูมิขนาดหลายมิติ โดยอนุภาค (Particle หรือ X) จะถูกวางไว้ในปริภูมิของคำตอบ (Solution Space) พร้อมกับความเร็วต้นค่าหนึ่งและช่องทางในการติดต่อกับอนุภาคอื่น [3, 4] อนุภาคจะเคลื่อนที่ไปเรื่อยๆ บนปริภูมิของคำตอบ ถ้าอนุภาคตัวใดเข้าใกล้คำตอบอนุภาคตัวนั้นก็จะมีค่าความสำคัญมากขึ้น และดึงดูดให้อนุภาคตัวอื่นๆ เบนเส้นทางเข้าหาอนุภาคตัวนั้น

สำหรับหลักการการทำงานของ PSO ได้แนวคิดมาจากการออกหาอาหารของฝูงสัตว์ เช่น ฝูงนก ฝูงปลา เป็นต้น โดยฝูงสัตว์เหล่านี้จะแต่งตั้งจำฝูงหรือหัวหน้าฝูง (Global BEST หรือ GBEST) ซึ่งเป็นตัวที่จะบินนำลูกฝูง (อนุภาค หรือ Particles) ไปหาอาหาร โดยลูกฝูงทุกตัวจะบินตามจำฝูง ซึ่งหลักการเหล่านี้ถูกนำมาประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุด โดยนกทุกตัวจะถูกสุ่มลงบนปริภูมิของคำตอบ โดยปริภูมิของคำตอบเปรียบเสมือนกับพื้นที่ขอบเขตที่นกต้องการหาอาหาร หากนกตัวใดพบแหล่งอาหารมากที่สุด นกตัวนั้นจะรับการแต่งตั้งให้เป็นจำฝูง ซึ่งแตกต่างจากธรรมชาติ ที่กำหนดว่าให้นกตัวที่แข็งแรงที่สุดเป็นจำฝูง แต่สิ่งที่เหมือนกันกับธรรมชาติ คือนกทุกตัวจะทำการเคลื่อนที่เข้าหานกที่เป็นจำฝูง และนกทุกตัวจะทำการค้นหาคำตอบบริเวณรอบๆ จำฝูงนั้นอย่างละเอียด โดยตั้งสมมติฐานว่า ในบริเวณที่มีแหล่งอาหารมาก จะมีโอกาสได้พบกับแหล่งอาหารที่มากขึ้นกว่าเดิม หรืออาจจะพิจารณาค้นหาคำตอบอย่างละเอียดเฉพาะในบริเวณที่ทราบว่ามีคำตอบที่ดีที่สุดที่สามารถหาพบ สำหรับบริเวณอื่น PSO จะค้นหาคำตอบแบบผ่านๆ ดังนั้นการค้นหาคำตอบจะไม่ค้นหาคำตอบทั้งหมดที่เป็นไปได้ แต่จะดำเนินการค้นหาคำตอบเฉพาะบริเวณใดบริเวณหนึ่งอย่างละเอียด จึงสามารถนำ PSO ไปประยุกต์ใช้กับปริภูมิของคำตอบขนาดใหญ่ได้

เมื่อเปรียบเทียบ PSO กับขั้นตอนวิธีเชิงวิวัฒนาการ (Evolutionary Algorithm หรือ EA) เช่น ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm หรือ GA) [5, 6] เป็นต้น PSO สามารถแก้ปัญหาการหาจุดที่ดีที่สุดได้ดีกว่าหลายปัญหา โดย PSO มีความเร็วในการลู่เข้าหาคำตอบมากกว่าและได้รับคำตอบจากการค้นหาที่ดีกว่า [7-9] สำหรับข้อดีของ PSO คือ สามารถเขียนโปรแกรมประยุกต์ใช้ได้ง่าย มีอัตราการลู่เข้าหาคำตอบเร็ว และมีพารามิเตอร์ (Parameters) ที่ต้องปรับค่าจำนวนน้อย แต่ PSO ก็มีข้อเสียคือ ปัญหาการลู่เร็วเกินไป (Premature Convergence Problem) หรือปัญหาการติดในจุดสุดสัมพัทธ์ (Local Optimum Problem) [10, 11] โดยปัญหานี้เมื่อเกิดขึ้นแล้วคำตอบที่ได้จากการค้นหาจะไม่ดีขึ้นไม่ว่าจะค้นหาต่อไปอีกยาวนานแค่ไหนก็ตาม

สาเหตุที่ PSO เกิดปัญหาการลู่เร็วเกินไปและปัญหาการติดในจุดสุดสัมพัทธ์ เนื่องมาจากหลักการทำงานของ PSO ที่เน้นการค้นหาคำตอบบริเวณรอบจำฝูง โดยค้นหาในขอบเขตที่แคบและไม่กระจายตัวออกไปค้นหาในบริเวณอื่นอีก ถ้าหากจำฝูงติดอยู่ในจุดสุดสัมพัทธ์ โดยหลักการทำงานของ PSO จะทำให้ลูกฝูงทั้งหมดเคลื่อนเข้าไปค้นหาคำตอบในบริเวณจุดสุดสัมพัทธ์ที่จำฝูงนั้นติดอยู่ ซึ่งการค้นหาคำตอบในบริเวณจุดสุดสัมพัทธ์จะทำให้คำตอบที่ได้นั้นไม่ดีขึ้น ไม่ว่าจะค้นหาคำตอบต่อไปอีกยาวนานแค่ไหนก็ตาม

จากผลกระทบของปัญหาการติดในจุดสุดสัมพัทธ์ ทำให้ผลลัพธ์การค้นหาคำตอบของ PSO ได้คำตอบที่ไม่ดีเท่าที่ควร ซึ่ง PSO จะมีประสิทธิภาพในการค้นหาคำตอบมากกว่านี้ ถ้าหากลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO ดังนั้นจึงเป็นเรื่องสำคัญที่จะพยายามลดปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO เพื่อให้ PSO ประสิทธิภาพในการค้นหาคำตอบเพิ่มขึ้น แต่สิ่งสำคัญอย่างหนึ่งคือ วิธีการที่นำไปแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO จะต้องไม่ทำให้ PSO เสียคุณสมบัติหลัก เช่น มีพารามิเตอร์มากเกินไปและขั้นตอนวิธีที่เพิ่มเข้าไปมีความซับซ้อนมากเกินไปทำให้นำไปประยุกต์ยากลำบาก ทำให้การค้นหาคำตอบของ PSO ในบางกรณีแย่งลงกว่าเดิม เป็นต้น

เพื่อแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO งานวิจัยจำนวนมาก [11-47] ได้เพิ่มความหลากหลายของประชากรเพื่อป้องกันอนุภาคลู่เข้าหาจุดสุดสัมพัทธ์ (Local Optimum) โดยเพิ่มเทคนิคการกลายพันธุ์ (Mutation Operators) เข้าไปในขั้นตอนการทำงานของ PSO แต่การกลายพันธุ์อาจจะทำให้เกิดปัญหาการลู่เร็วเกินไปมากขึ้นเมื่อเปรียบเทียบกับ PSO ปกติ ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้ทำการศึกษาและปรับปรุงการกลายพันธุ์ที่นำไปประยุกต์ใช้กับ PSO เพื่อแก้ปัญหาการติดในจุดสุดสัมพัทธ์โดยไม่เพิ่มปัญหาการลู่เร็วเกินไปเมื่อเปรียบเทียบกับ PSO ปกติ

โดยการกลายพันธุ์จะเป็นการเพิ่มขอบเขตการค้นหาของ PSO ให้กว้างขึ้น ให้เพียงพอที่จะครอบคลุมตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ที่จำฝูงนั้นติดอยู่ ทำให้มีโอกาสที่จะพบกับคำตอบที่ดีกว่าจุดสุดสัมพัทธ์ที่จำฝูงนั้นติดอยู่ และหลุดออกจากจุดสุดสัมพัทธ์ที่ติดอยู่นั้น โดยการเพิ่มขอบเขตการค้นหาคำตอบสามารถสังเกตได้จากความหลากหลายของประชากร โดยถ้าความหลากหลายของประชากรมาก ขอบเขตการค้นหาคำตอบก็จะมาก ในทางกลับกันถ้าความหลากหลายของประชากรน้อย ขอบเขตการค้นหาคำตอบก็จะน้อย

โดยส่วนมากแล้วการนำการกลายพันธุ์มาประยุกต์ใช้กับ PSO มักประสบปัญหาเรื่องความยุ่งยากในการปรับพารามิเตอร์ [32] ซึ่งเป็นเหตุผลที่ทำให้การนำไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดโดยทั่วไปนั้นเป็นเรื่องที่ยากลำบาก เพราะต้องปรับพารามิเตอร์ของการกลายพันธุ์ให้เหมาะสมกับปัญหาการหาจุดที่ดีที่สุดที่จะดำเนินการแก้ไข ไม่เช่นนั้นผลลัพธ์ในการค้นหาคำตอบจากการนำการกลายพันธุ์มาประยุกต์ใช้กับ PSO จะให้ผลลัพธ์ที่แย่กว่า PSO ที่ไม่ดำเนินการกลายพันธุ์

จากเหตุนี้จึงเป็นสาเหตุให้งานวิจัยนี้ดำเนินการศึกษาค้นคว้าพัฒนาการกลายพันธุ์ที่นำไปประยุกต์ใช้กับ PSO ให้มีความยุ่งยากในการปรับพารามิเตอร์น้อยลง โดยให้การกลายพันธุ์นั้นไม่ส่งผลกระทบต่อการทำงานของ PSO และช่วยลดข้อด้อยของ PSO นั่นคือปัญหาการติดในจุดสุดสัมพัทธ์ โดยการเพิ่มความหลากหลายของประชากร ทำให้การกลายพันธุ์ที่นำเสนอนี้สามารถนำไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดได้ง่ายขึ้น และได้รับคำตอบที่ดีขึ้นเมื่อเปรียบเทียบกับ PSO ที่ไม่ดำเนินการกลายพันธุ์

ดังนั้นวิธีการกลายพันธุ์ที่นำเสนอนี้จะเป็นการลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO เนื่องจากวิธีการกลายพันธุ์ที่นำเสนอนี้ได้เพิ่มความหลากหลายให้กับประชากรของ PSO นอกจากนี้วิธีการกลายพันธุ์ที่นำเสนอนี้ยังช่วยลดความยุ่งยากในการปรับพารามิเตอร์ทำให้สามารถนำไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดได้ง่าย จากเหตุผลที่กล่าวมานี้ทำให้วิธีการกลายพันธุ์ที่นำเสนอนี้ได้รับคำตอบที่ดีกว่าเมื่อเปรียบเทียบกับ PSO

ถึงแม้การกลายพันธุ์จะช่วยให้ขอบเขตการค้นหาคำตอบของ PSO เพิ่มขึ้น ซึ่งช่วยลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ แต่การกลายพันธุ์ก็เพิ่มขอบเขตให้ PSO ได้จำกัด ถ้าหากขอบเขตของการกลายพันธุ์ไม่มากพอที่จะครอบคลุมตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ที่จำฝูงติดอยู่ การนำการกลายพันธุ์มาประยุกต์ใช้ก็อาจจะไม่สามารถช่วยแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO เรียกปัญหานี้ว่า ปัญหาขอบเขตการกลายพันธุ์

เพื่อแก้ปัญหาขอบเขตการกลายพันธุ์ มีงานวิจัย [48-52] ที่เพิ่มวิธีการเริ่มต้นการค้นหาใหม่หรือรีรัน (RERUN) หรือรีเซ็ต (Reset) เข้าไปในขั้นตอนการทำงานของ PSO โดยวิธีรีรันจะดำเนินการเมื่อเกิดปัญหาขอบเขตการกลายพันธุ์สำหรับ PSO ที่ประยุกต์การกลายพันธุ์และปัญหาการติดในจุดสุดสัมพัทธ์สำหรับ PSO ปกติ สำหรับวิธีรีรันทำให้อนุภาคกระจายตัวออกเป็นบริเวณกว้างกว่าการกระจายตัวจากการกลายพันธุ์ จึงมีขอบเขตการค้นหาที่กว้างกว่าการกลายพันธุ์ ดังนั้นอนุภาคจึงมีโอกาสที่กระจายตัวออกไปนอกขอบเขตที่การกลายพันธุ์ไม่สามารถกระจายตัวไปถึง แต่วิธีรีรันก็มีข้อเสียคืออนุภาคจะถูกกำหนดค่าเริ่มต้นใหม่ อนุภาคไม่สามารถกำหนดขอบเขตการกระจายตัวได้ ดังนั้นอนุภาคกระจายตัวเป็นบริเวณที่กว้างมากเกินไป ผลที่ตามก็คืออนุภาคต้องใช้เวลานานเพื่อให้อนุภาคกลับเข้ามาหากันใหม่หลังการดำเนินการรีรัน ซึ่งทำให้สูญเสียเวลาในการค้นหาคำตอบและลดโอกาสในการลองค้นหาคำตอบโดยไม่จำเป็น

จากเหตุนี้จึงเป็นสาเหตุให้งานวิจัยนี้ดำเนินการศึกษาค้นคว้าพัฒนาและปรับปรุงรีรัน เพื่อลดระยะเวลาในการลู่เข้าหากันหลังจากการรีเซ็ต โดยทำให้รีรันสามารถกำหนดขอบเขตการค้นหาคำตอบได้ เพื่อให้อนุภาคกระจายตัวออกจากกันเท่าที่จำเป็น ซึ่งทำให้โอกาสในการลองค้นหาคำตอบนั้นเพิ่มขึ้น ส่งผลให้คำตอบที่ได้รับจากการค้นหานั้นดีขึ้น แต่ขอบเขตที่อนุภาคกระจายตัวออกจากกันนั้นจะมีขอบเขตที่มากกว่าการกลายพันธุ์มาก เพื่อให้อนุภาคออกไปค้นหาคำตอบในบริเวณอื่นๆ นอกจากจุดสุดสัมพัทธ์ที่จำฝูงติดอยู่ จึงเป็นการแก้ปัญหาการติดในจุดสุดสัมพัทธ์และปัญหาขอบเขตการกลายพันธุ์ โดยเรียกวิธีใหม่นี้ว่าการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิมหรือรีโพซิชัน (REPOSITION) หรือรีเซ็ต

เนื่องจากหลักการทำงานของ PSO คือ ฝูงจะต้องลู่เข้าหาจำฝูง ดังนั้นเมื่อกระจายอนุภาคออกจากกันด้วยรีโพซิชัน ด้วยหลักการทำงานของ PSO อนุภาคก็จะลู่เข้ามาค้นหาคำตอบในบริเวณจุดสุดสัมพัทธ์ที่จำฝูงติดอยู่หรือบริเวณเดิมก่อนการรีเซ็ต ซึ่งผิดจุดประสงค์ของการดำเนินการรีโพซิชันที่ต้องการให้อนุภาคกระจายตัวออกไปค้นหาในบริเวณอื่น โดยการกระจายอนุภาคออกไปค้นหาคำตอบในบริเวณอื่นนั้นจะทำให้ผลลัพธ์จากการค้นหาคำตอบดีกว่าที่จะให้อนุภาคเคลื่อนที่กลับเข้าไปค้นหาในบริเวณเดิม

จากเหตุนี้จึงเป็นสาเหตุให้งานวิจัยนี้ดำเนินการศึกษาค้นคว้าพัฒนาและปรับปรุงรีโพซิชัน ให้มีประสิทธิภาพการค้นหาคำตอบที่ดีขึ้นโดยหลังจากการรีโพซิชันแล้วอนุภาคสามารถที่จะลู่เข้าไปค้นหาในบริเวณใหม่ซึ่งจะทำให้ผลลัพธ์จากการค้นหาคำตอบนั้นดีขึ้น

สำหรับ PSO สามารถนำมาประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดได้อย่างมากมาย โดยมีงานวิจัยจำนวนมากมายได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดทั้งในงานด้านการขนส่ง วิทยาศาสตร์ วิศวกรรม ธุรกิจ และอุตสาหกรรม เป็นต้น สำหรับตัวอย่างงานวิจัยที่ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุด เช่น งานวิจัย [119, 120] ได้นำ PSO มาประยุกต์ใช้ฝึกโครงข่ายประสาท (Neural Network) และปรับพารามิเตอร์ของโครงข่ายประสาท สำหรับงานวิจัย [121 - 123] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการแบ่งกลุ่มข้อมูล (Data Clustering) สำหรับงานวิจัย [124, 125] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาเรื่องขนส่ง (Transportation Problem) สำหรับงานวิจัย [126] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการจัดทรัพยากรของการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) สำหรับงานวิจัย [127] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการจัดเครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensor Network) สำหรับงานวิจัย [128 - 130] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการปรับค่าพารามิเตอร์ที่ใช้ในจัดการระบบไฟฟ้า (Power Systems) สำหรับงานวิจัย [131 - 133] ได้นำหลักการค้นหาของ PSO มาประยุกต์ใช้การค้นหาของหุ่นยนต์ (Robot Search) สำหรับงานวิจัย [134] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการจัดการระบบสื่อสาร (Transmission Network) สำหรับงานวิจัย [135 - 137] ได้นำมาประยุกต์ใช้การกำหนดรายการ (Scheduling) เป็นต้น แต่เนื่องจาก PSO ประสบปัญหาการติดในจุดสุดสัมพัทธ์ จึงทำให้ผลลัพธ์ที่ได้จากการค้นหาคำตอบไม่ดีเท่าที่ควร เมื่อนำไปประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดเหล่านี้จึงได้ผลลัพธ์จากการค้นหาคำตอบไม่ดีเท่าที่ควร ซึ่งถ้าหากปรับปรุง PSO ให้ลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์จะทำให้การนำ PSO ไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดเหล่านี้ ได้ผลลัพธ์จากการค้นหาคำตอบที่ดีขึ้น ดังนั้นการแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO ก็เท่ากับเป็นการเพิ่มประสิทธิภาพในการนำ PSO ไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดเหล่านี้ โดยทำให้ได้รับผลลัพธ์จากการค้นหาคำตอบดีขึ้นจึงเป็นที่มาของงานวิจัยนี้ที่จะมุ่งเน้นลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO โดยการนำวิธีการกลายพันธุ์และรีโพอิชันมาประยุกต์ใช้ เพื่อให้ PSO ไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดเหล่านี้ให้ได้ผลลัพธ์การค้นหาที่ดียิ่งขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังศึกษาแนวทางปรับปรุงประสิทธิภาพการค้นหาของ PSO โดยนำวิธีการกลายพันธุ์และรีโพอิชันมาประยุกต์ใช้ในการแก้ไขหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ เพื่อให้คำตอบที่ได้จากการค้นหานั้นดีขึ้น โดยทำการทดลองกับปัญหาฟังก์ชันมาตรฐาน (Benchmark Test Functions) ปัญหาถุงเป้หลายมิติ (Multidimensional Knapsack Problems หรือ MKP) และปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem หรือ TSP)

1.3 สมมติฐานของการศึกษา

โดยปกติการค้นหาคำตอบจากปัญหาการหาจุดที่ดีที่สุด จะไม่สามารถทราบคำตอบที่พบเป็นคำตอบที่ดีที่สุด สำหรับงานวิจัยนี้ได้ดำเนินการศึกษาและปรับปรุง PSO ให้มีประสิทธิภาพการค้นหาและได้รับคำตอบที่ดีขึ้น แต่ไม่รับประกันว่าคำตอบที่ได้จากการค้นหานั้นเป็นคำตอบที่ดีที่สุด โดยการเปรียบเทียบขั้นตอนวิธีว่า ขั้นตอนวิธีใดมีประสิทธิภาพการค้นหาดีกว่ากันซึ่งจะเปรียบเทียบ

จากคำตอบที่ได้จากการค้นหา โดยขั้นตอนวิธีที่สามารถหาคำตอบได้ดีกว่า ขั้นตอนวิธีนั้นจะมีประสิทธิภาพการค้นหาที่ดีกว่า แต่สำหรับกรณีที่อยู่คำตอบที่ดีที่สุดและขั้นตอนวิธีสามารถหาคำตอบที่ดีที่สุดพบ จะใช้วิธีการเปรียบเทียบว่าขั้นตอนวิธีที่มีจำนวนพบคำตอบที่ดีที่สุดมากกว่าเป็นขั้นตอนวิธีที่มีประสิทธิภาพการค้นหาที่ดีกว่า แต่สำหรับกรณีที่อยู่คำตอบที่ดีที่สุดและขั้นตอนวิธีไม่สามารถหาคำตอบที่ดีที่สุดพบ จะใช้วิธีการเปรียบเทียบว่า ขั้นตอนวิธีที่สามารถหาคำตอบได้ดีกว่าเป็นขั้นตอนวิธีที่มีประสิทธิภาพในการค้นหาที่ดีกว่า เพราะขั้นตอนวิธีที่เปรียบเทียบนั้นเกิดปัญหาการติดในจุดสุดสัมพัทธ์ แต่ขั้นตอนวิธีที่สามารถติดจุดสุดสัมพัทธ์ในตำแหน่งที่ดีกว่าเป็นขั้นตอนวิธีที่มีประสิทธิภาพการค้นหาที่ดีกว่า

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

แนวคิดหลักของวิทยานิพนธ์ฉบับนี้ คือความพยายามที่จะแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO โดยใช้วิธีการกลายพันธุ์และรีโพอิชัน จากสมมติฐานที่ว่า PSO มีข้อเสียหลัก คือปัญหาการติดในจุดสุดสัมพัทธ์ ซึ่งสามารถแก้ไขหรือลดปัญหานี้ได้ โดยนำการกลายพันธุ์และรีโพอิชันมาประยุกต์ใช้กับ PSO สำหรับการกลายพันธุ์อาจจะส่งผลให้เกิดปัญหาใหม่ คือปัญหาการลู่อูเร็วเกินไปที่รุนแรงมากขึ้นเมื่อเปรียบเทียบกับ PSO ที่ไม่ดำเนินการกลายพันธุ์ ซึ่งปัญหานี้สามารถแก้ไขหรือลดได้โดยออกแบบขั้นตอนวิธีให้ความเร็วในการลู่อูเข้าหากันของอนุภาคเทียบเท่ากับ PSO ที่ไม่ดำเนินการกลายพันธุ์ ซึ่งจะหาคำตอบที่ได้จากการค้นหานั้นดีขึ้น อย่างไรก็ตามการประยุกต์ใช้การกลายพันธุ์เพียงอย่างเดียวก็มีโอกาสประสบปัญหาการติดในจุดสุดสัมพัทธ์ ดังนั้นจึงนำรีโพอิชันมาประยุกต์ใช้เพื่อแก้ปัญหาการติดในจุดสุดสัมพัทธ์ที่เกิดขึ้นกับการกลายพันธุ์ สำหรับรีโพอิชันถ้าหากไม่มีการรีเซ็ตตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัว (PBEST) และตำแหน่งที่ดีที่สุดของอนุภาคทั้งหมด (GBEST) อนุภาคจะเคลื่อนที่กลับมาบริเวณเดิม จึงหาคำตอบที่ได้จากการค้นหาไม่ดีเท่าที่ควร เพื่อให้อนุภาคเคลื่อนที่ไปค้นหาในบริเวณใหม่จึงต้องรีเซ็ต PBEST และ GBEST ซึ่งหาคำตอบที่ได้นั้นดีขึ้น นอกจากนี้การประยุกต์ใช้รีโพอิชันกับ PSO เพียงอย่างเดียว อาจจะทำให้ประชากรขาดความหลากหลาย จึงหาคำตอบที่ได้จากการค้นหานั้นไม่ดีเท่าที่ควร เพื่อแก้ไขข้อเสียของกันและกันจึงควรที่จะนำทั้งสองวิธีมาประยุกต์ใช้ร่วมกันจึงจะทำให้การค้นหานั้นได้รับผลลัพธ์ที่ดีที่สุด

1.5 ขอบเขตการวิจัย

วิทยานิพนธ์ฉบับนี้ทำการศึกษาและพัฒนาการประยุกต์การกลายพันธุ์และรีโพอิชันกับ PSO โดยกำหนดขอบเขตของงานวิจัยดังนี้

1. งานวิจัยนี้เน้นการปรับปรุง PSO ให้การค้นหาได้รับคำตอบที่ดีขึ้น โดยพยายามที่จะแก้ไขปัญหาการติดในจุดสุดสัมพัทธ์ ซึ่งเป็นข้อเสียหลักของ PSO โดยขั้นตอนวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอสามารถทำให้หลุดจากจุดสุดสัมพัทธ์เดิมที่ PSO ประสบปัญหาแล้วไปพบจุดสุดสัมพัทธ์ใหม่ที่ดีกว่าเดิมหรืออาจจะพบจุดสุดสัมบูรณ์ (Global Optimum) ดังนั้นขั้นตอนวิธีที่นำเสนอนี้จะไม่สามารถหาคำตอบที่ดีที่สุดของปัญหาที่นำมาแก้ไข แต่ขั้นตอนวิธีที่นำเสนอนี้จะได้รับคำตอบจากค้นหาที่ดีกว่า PSO

2. งานวิจัยนี้เน้นการปรับปรุง PSO ให้สามารถค้นหาได้รับคำตอบที่ดีกว่า PSO ทั้งในกรณีปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่อง (Continuous Optimization Problem) และปัญหาการหาจุดที่ดีที่สุดเชิงการตัดสินใจ (Discrete Optimization Problem)
3. เพื่อวัดประสิทธิภาพของวิธีการที่นำเสนอในวิทยานิพนธ์ฉบับนี้ ได้แบ่งการทดลองออกเป็น 3 ส่วน ส่วนที่หนึ่งทำทดลองกับปัญหาฟังก์ชันมาตรฐาน (Benchmark Test Functions) [54-60, 100, 101, 113] ส่วนที่สองทำทดลองกับปัญหาถุงเป้หลายมิติ (Multidimensional Knapsack Problems หรือ MKP) [61] ส่วนที่สามทำทดลองกับปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem หรือ TSP) [62]

1.6 ขั้นตอนการศึกษา

1. ศึกษาทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องกับ PSO
2. ศึกษาทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องกับปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ และปัญหาการเดินทางของพนักงานขาย
3. ศึกษาการวิธีการประยุกต์การกลายพันธุ์กับ PSO
4. ศึกษาผลกระทบจากวิธีการประยุกต์การกลายพันธุ์กับ PSO
5. ศึกษาแนวทางการแก้ปัญหาที่เกิดจากการประยุกต์การกลายพันธุ์กับ PSO
6. ศึกษาการวิธีการประยุกต์รีรันและรีโพซิชันกับ PSO
7. ศึกษาผลกระทบจากวิธีการประยุกต์รีรันและรีโพซิชันกับ PSO
8. ศึกษาแนวทางการแก้ปัญหาที่เกิดจากการประยุกต์รีโพซิชันกับ PSO
9. ทดลองเพื่อเปรียบเทียบผลลัพธ์จากปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ และปัญหาการเดินทางของพนักงานขาย เพื่อศึกษาข้อดีและข้อเสียของวิธีการกลายพันธุ์และรีโพซิชัน
10. พัฒนาวิธีการกลายพันธุ์และรีโพซิชัน
11. สรุปผลการทำงานของวิธีการกลายพันธุ์และรีโพซิชันที่พัฒนาขึ้นพร้อมวิเคราะห์ผลที่ได้
12. จัดทำเอกสารประกอบวิทยานิพนธ์

1.7 เครื่องมือและอุปกรณ์ที่ใช้ในงานวิจัย

1. เครื่องคอมพิวเตอร์ส่วนบุคคล
 - หน่วยประมวลผลกลางยี่ห้อ Intel รุ่น Core i7 3770 ความเร็ว 2.4 GHz
 - หน่วยความจำหลักขนาด 8 GB
2. ระบบปฏิบัติการ Windows Seven Professional
3. โปรแกรม Microsoft Visual C++ 2005

1.8 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 6 บท และภาคผนวก 9 บท ดังนี้

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ขอบเขตของการวิจัยและขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่เกี่ยวข้อง คือ ปัญหาการหาจุดที่ดีที่สุด การหาจุดที่ดีที่สุด ปัญหาการติดในจุดสุดสัมพัทธ์ วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค ตัวอย่างการประยุกต์ใช้ PSO ในทางปฏิบัติ ข้อดีและข้อเสียของ PSO

บทที่ 3 กล่าวถึงงานวิจัยที่เกี่ยวข้อง คือ การประยุกต์การกลายพันธุ์ รีรัน และรีโพอิชชันกับ PSO

บทที่ 4 กล่าวถึงการปรับปรุงการกลายพันธุ์และรีโพอิชชัน ซึ่งเป็นการอธิบายแนวคิดและที่มาของวิธีการที่วิทยานิพนธ์ฉบับนี้นำเสนอ โดยการเพิ่มประสิทธิภาพของ PSO ที่ประยุกต์ใช้การกลายพันธุ์ ประสิทธิภาพการเปรียบเทียบโดยลดปัญหาการลู่เร็วเกินไปและการเพิ่มของประสิทธิภาพของ PSO หลังจากเกิดปัญหาการติดในจุดสุดสัมพัทธ์ และวิเคราะห์ข้อดีและข้อเสียของวิธีที่นำเสนอ

บทที่ 5 กล่าวถึงการทดลองเพื่ออ้างอิงแนวคิดและที่มาของวิธีการที่วิทยานิพนธ์ฉบับนี้นำเสนอ และเปรียบเทียบผลลัพธ์กับ PSO วิธีการอื่นที่มีลักษณะใกล้เคียงกับวิธีการที่นำเสนอ โดยดำเนินการทดลองกับปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ และปัญหาการเดินทางของพนักงานขาย

บทที่ 6 กล่าวถึงบทสรุปผลการทดลองและข้อเสนอแนะ รวมทั้งแนวทางในการพัฒนาโปรแกรม

ภาคผนวก ก รายละเอียดของฟังก์ชันมาตรฐาน ที่ใช้ในการทดลองในบทที่ 5

ภาคผนวก ข การประยุกต์ PSO กับปัญหาถุงเป้แบบหลายมิติ การประยุกต์การกลายพันธุ์ประเภทไม่มีการเปรียบเทียบและรีโพอิชชันกับ PSO กับปัญหาถุงเป้แบบหลายมิติ การประยุกต์การกลายพันธุ์ประเภทมีการเปรียบเทียบกับ PSO กับปัญหาถุงเป้แบบหลายมิติ

ภาคผนวก ค การประยุกต์ PSO กับปัญหาการเดินทางของพนักงานขาย การประยุกต์การกลายพันธุ์ประเภทไม่มีการเปรียบเทียบและรีโพอิชชันกับ PSO กับปัญหาการเดินทางของพนักงานขาย การประยุกต์การกลายพันธุ์ประเภทมีการเปรียบเทียบกับ PSO กับปัญหาการเดินทางของพนักงานขาย

ภาคผนวก ง กล่าวถึงการทดลองเพื่ออ้างอิงแนวคิดและที่มาของวิธีการที่วิทยานิพนธ์ฉบับนี้นำเสนอโดยทำการทดลองกับปัญหาถุงเป้หลายมิติ

ภาคผนวก จ กล่าวถึงการทดลองเพื่ออ้างอิงแนวคิดและที่มาของวิธีการที่วิทยานิพนธ์ฉบับนี้นำเสนอโดยทำการทดลองกับปัญหาการเดินทางของพนักงานขาย

ภาคผนวก ฉ รหัสจำลองของขั้นตอนวิธีการกลายพันธุ์และรีโพอิชชันที่ใช้ในการทดลองในบทที่ 5

ภาคผนวก ช ผลงานวิจัยที่ได้รับการตีพิมพ์

ภาคผนวก ซ การทดลองการปรับพารามิเตอร์ของการกลายพันธุ์และรีโพอิชชัน

ภาคผนวก ฌ การเปรียบเทียบการกลายพันธุ์และรีโพอิชชันกับขั้นตอนวิธีอื่นที่มีลักษณะใกล้เคียงกับขั้นตอนวิธีการที่นำเสนอ เพื่อแสดงให้เห็นว่าขั้นตอนวิธีการที่นำเสนอนั้นสามารถเพิ่มประสิทธิภาพการค้นหาของ PSO ได้อย่างมีประสิทธิภาพมากที่สุดเมื่อเปรียบเทียบกับขั้นตอนวิธีอื่นที่มีลักษณะใกล้เคียง

บทที่ 2

ทฤษฎีพื้นฐานที่เกี่ยวข้อง

2.1 ปัญหาการหาจุดที่ดีที่สุด

2.1.1 นิยามของปัญหาการหาจุดที่ดีที่สุด

2.1.1.1 ปัญหาการหาจุดที่ดีที่สุด

ปัญหาการหาจุดที่ดีที่สุด (Optimization Problems) คือปัญหาเกิดจากการพยายามที่จะค้นหาคำตอบที่ดีที่สุดจากคำตอบที่เป็นไปได้ทั้งหมด [63, 64] โดยปัญหานี้สามารถแสดงในรูปของสมการทางคณิตศาสตร์ได้ดังนี้ [63, 66-69]

$$\begin{aligned} \text{Min or Max. } & f(x) \quad x \in R, Z \\ \text{Subject } & g_i(x) \leq a_i, \quad i = 1, \dots, m \\ & h_i(x) = b_i, \quad i = 1, \dots, p \end{aligned} \quad (2.1)$$

โดยที่ x เป็นตัวแปรต้น $f(x)$ เป็นฟังก์ชันเป้าหมาย $g_i(x)$ เป็นข้อจำกัดที่ไม่เท่ากัน (Inequality Constraints) $h_i(x)$ เป็นข้อจำกัดที่เท่ากัน (Equality Constraints) a_i และ b_i เป็นค่าคงที่ซึ่งเป็นตัวกำหนดขอบเขตของปัญหาการหาจุดที่ดีที่สุด R คือจำนวนจริง Z คือจำนวนเต็ม

2.1.1.2 ฟังก์ชันเป้าหมาย

ฟังก์ชันเป้าหมาย (Objective function) หรือฟังก์ชันต้นทุน (Cost function) คือฟังก์ชันที่นำมาใช้เป็นตัวแทนของปัญหาการหาจุดที่ดีที่สุดเนื่องจากคอมพิวเตอร์ไม่สามารถแก้ไขปัญหาการหาจุดที่ดีที่สุดได้โดยตรง จึงจำเป็นต้องแปลงปัญหาการหาจุดที่ดีที่สุดในรูปแบบของฟังก์ชันทางคณิตศาสตร์ คอมพิวเตอร์จึงสามารถแก้ไขปัญหาการหาจุดที่ดีที่สุดได้ ดังนั้นคำตอบที่ได้จากการแก้ปัญหของฟังก์ชันเป้าหมาย คือคำตอบที่ได้จากการแก้ปัญหาการหาจุดที่ดีที่สุด [65]

2.1.1.3 ตัวแปรต้น

ตัวแปรต้น (Control Variables) คือตัวแปรตามที่อยู่ในฟังก์ชันเป้าหมาย เช่น $f(x_1, x_2, \dots, x_n)$ โดยตัวแปรต้นจะถูกแทนค่า เพื่อให้ฟังก์ชันเป้าหมายคำนวณออกมาได้ค่าความเหมาะสม (Fitness) มีค่ามากที่สุดหรือค่าน้อยที่สุด [65]

2.1.1.4 ข้อจำกัด

ข้อจำกัด (Constraint) คือขอบเขตหรือเงื่อนไขสำหรับการหาค่าตัวแปรต้นที่อยู่ในฟังก์ชันเป้าหมาย [65, 66] โดยเงื่อนไขเหล่านี้ อาจเกิดมาจากการหาจุดที่ดีที่สุดหรือคุณสมบัติของตัวแปรต้น เช่น ถ้ากำหนดให้ตัวแปรต้น คือ น้ำหนัก ระยะทาง เวลา ตัวแปรต้นนั้นจะมีเงื่อนไขเพิ่มขึ้นมา คือต้องมีค่ามากกว่าหรือเท่ากับศูนย์ เพราะน้ำหนัก ระยะทาง เวลา ไม่มีโอกาสติดลบ เป็นต้น

2.1.2 ประเภทของปัญหาการหาจุดที่ดีที่สุด

2.1.2.1 ปัญหาการหาจุดที่ดีที่สุดที่ต้องการหาค่าสูงสุดและค่าต่ำสุด

ปัญหาการหาจุดที่ดีที่สุดที่ต้องการหาค่าสูงสุดและค่าต่ำสุด [65, 66] เป็นการแบ่งปัญหาการหาจุดที่ดีที่สุดโดยพิจารณาจากเป้าหมายของการค้นหา ซึ่งสามารถแบ่งออกได้ 2 กลุ่ม คือปัญหาที่ต้องการหาค่าสูงสุด (Maximum Optimization Problem) เช่น ต้องการจำนวนลูกค้าเข้ามารับบริการในธนาคารต่อชั่วโมงให้ได้มากที่สุด ต้องการให้จำนวนไมล์ที่รถวิ่งไปได้ต่อน้ำมัน 1 แกลลอนของรถชนิดหนึ่งให้ได้มากที่สุด ฯลฯ และปัญหาที่ต้องการหาค่าต่ำสุด (Minimum Optimization Problem) เช่น ต้องการให้ต้นทุนต่อหน่วยในการผลิตสินค้าให้น้อยที่สุด ต้องการให้เวลาที่ใช้ในการผลิตสินค้าต่อชิ้นน้อยที่สุด ฯลฯ

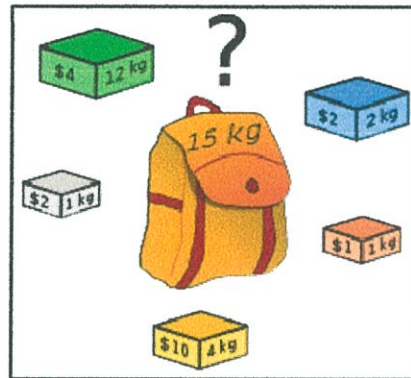
2.1.2.2 ปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่องและเชิงตัดสินใจ

ปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่องและเชิงตัดสินใจ [64, 66, 68] เป็นการแบ่งปัญหาการหาจุดที่ดีที่สุดโดยพิจารณาจากตัวแปรต้นของปัญหา ซึ่งสามารถแบ่งออกได้ 2 กลุ่ม คือปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่อง (Continuous Optimization Problem) และปัญหาการหาจุดที่ดีที่สุดเชิงการตัดสินใจ (Discrete Optimization Problem หรือ Combinatorial Optimization Problem)

ปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่อง คือปัญหาการหาจุดที่ดีที่สุดที่มีตัวแปรต้นที่มีค่าเป็นจำนวนจริง โดยค่าของตัวแปรต้นจะมีค่าอยู่ในช่วงใดช่วงหนึ่งที่มีค่าต่อเนื่องเป็นจำนวนจริง [68-70] ตัวอย่างของปัญหานี้เช่น ปัญหาการหาค่าสูงสุดหรือค่าต่ำสุดของฟังก์ชันมาตรฐาน (Benchmark Test Functions) เป็นต้น

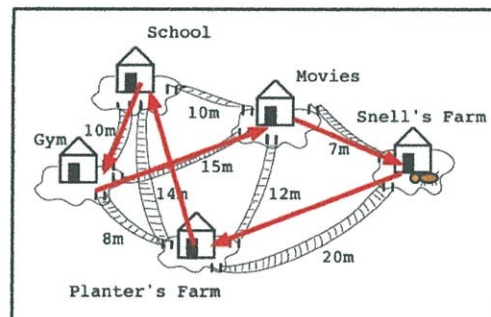
ปัญหาการหาจุดที่ดีที่สุดเชิงการตัดสินใจ คือปัญหาการหาจุดที่ดีที่สุดที่สามารถมีตัวแปรต้นเป็นค่าเหล่านี้ ไบนารี (Binary) ซึ่งถูกจำกัดให้มีค่าได้เพียงแค่ 0 กับ 1 เท่านั้น จำนวนเต็ม (Integers) เซตของวัตถุที่มีจำนวนมาก (Sets) การสลับลำดับกันในพีชคณิต (Permutations) กราฟ (Graphs) โดยตัวแปรต้นนั้นจะถูกกำหนดขอบเขตให้มีค่าอยู่ในช่วงใดช่วงหนึ่ง [66, 68] ตัวอย่างของปัญหานี้เช่น ปัญหาถุงเป้ (Knapsack Problem หรือ Rucksack Problem หรือ KP) และปัญหาการเดินทางของพนักงานขาย (Traveling Salesperson problem หรือ TSP) เป็นต้น

ปัญหาถุงเป้ (Knapsack Problem) [74] คือปัญหาการเลือกวัตถุใส่ลงไปในถุงเป้เพื่อให้มีมูลค่ารวมมากที่สุดโดยที่ถุงเป้ไม่ขาด โดยลักษณะของปัญหามีรายละเอียดดังนี้ มีวัตถุอยู่จำนวน n ชิ้น ซึ่งมีราคาเป็น v_1, v_2, \dots, v_n และมีน้ำหนักเป็น w_1, w_2, \dots, w_n และถุงเป้สามารถรับน้ำหนักได้มากที่สุดเท่ากับ W ให้เลือกวัตถุให้ได้มูลค่ารวมมากที่สุดโดยที่ถุงเป้สามารถรับน้ำหนักได้ เช่น มีวัตถุอยู่ 5 ชิ้น โดยแต่ละชิ้นไม่สามารถแบ่งแยกชิ้นส่วน ชิ้นแรกหนัก 12 กิโลกรัม มีมูลค่า 4 บาท ชิ้นที่สองหนัก 1 กิโลกรัม มีมูลค่า 2 บาท ชิ้นที่สามหนัก 2 กิโลกรัม มีมูลค่า 2 บาท ชิ้นที่สี่หนัก 4 กิโลกรัม มีมูลค่า 10 บาท และชิ้นที่ห้าหนัก 1 กิโลกรัม มีมูลค่า 1 บาท และถุงเป้สามารถใส่ได้มากที่สุด 15 กิโลกรัม ต้องการเลือกวัตถุให้มีมูลค่ารวมกันแล้วมากที่สุดโดยที่ถุงเป้ไม่ขาด ดังรูปข้างล่างนี้



รูปที่ 2.1 ตัวอย่างปัญหาถุงเป้ [74]

ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem หรือ TSP) [75] คือพนักงานขายต้องการหาเส้นทางการเดินทางไปยังลูกค้าทุกคนด้วยระยะทางที่น้อยที่สุด โดยที่ลูกค้าแต่ละรายจะได้รับบริการจากพนักงานขายเพียงคนละครั้งเดียวเท่านั้น ดังรูปข้างล่างนี้

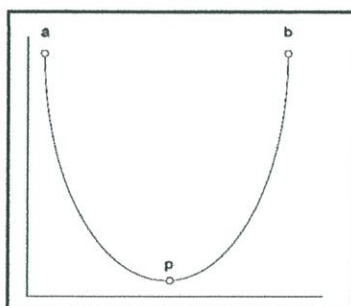


รูปที่ 2.2 ตัวอย่างปัญหาการเดินทางของพนักงานขาย [78]

2.1.2.3 ปัญหาการหาจุดที่ดีที่สุดฐานนิยมเดียวและหลายรูปแบบ

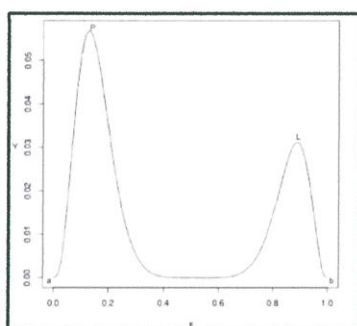
ปัญหาการหาจุดที่ดีที่สุดฐานนิยมเดียวและหลายรูปแบบเป็นการแบ่งปัญหาการหาจุดที่ดีที่สุดโดยพิจารณาจากคุณสมบัติของฟังก์ชันเป้าหมาย ซึ่งสามารถแบ่งออกได้ 2 กลุ่ม [75] คือปัญหาฐานนิยมเดียวหรือปัญหายูนิโมดอล (Unimodal Problem) และปัญหาหลายรูปแบบหรือปัญหาพหุโมดอล (Multimodal Problem)

ปัญหาหาค่าเหมาะที่สุดที่ไม่ใช่เชิงเส้น คือปัญหาการหาจุดที่ดีที่สุดที่ในขอบเขตการค้นหาไม่มีจุดสุดสัมพัทธ์ (Local Optimum) เลยและมีเพียงจุดสุดสัมบูรณ์ (Global Optimum) เพียงจุดเดียวเท่านั้น [71] ตัวอย่างของปัญหาดังรูปข้างล่างนี้ โดยกำหนดฟังก์ชัน $f(x)$ ให้มีขอบเขตอยู่ในช่วงระหว่าง a ถึง b จากรูปมีจุด P เพียงจุดเดียวเท่านั้นที่มีค่าน้อยที่สุดในขอบเขตระหว่าง a ถึง b ซึ่งเรียกจุด P ว่าจุดสุดสัมบูรณ์



รูปที่ 2.3 ตัวอย่างปัญหาหาค่าเหมาะที่สุดที่ไม่ใช่เชิงเส้น

ปัญหาหาค่าเหมาะที่สุดที่มีหลายจุดที่ดีที่สุด คือปัญหาการหาจุดที่ดีที่สุดที่ในขอบเขตการค้นหามีจุดสุดสัมพัทธ์ตั้งแต่หนึ่งจุดขึ้นไป แต่มีจุดสุดสัมบูรณ์เพียงจุดเดียวเท่านั้น [71] ตัวอย่างปัญหาดังรูปข้างล่างนี้ กำหนดฟังก์ชัน $f(x)$ ให้มีขอบเขตอยู่ในช่วงระหว่าง a ถึง b จากรูปมีจุด P เพียงจุดเดียวเท่านั้นที่มีค่ามากที่สุดภายในขอบเขตอยู่ในช่วงระหว่าง a ถึง b ซึ่งเรียกจุด P ว่าจุดสูงสุดสัมบูรณ์และมีจุด L เป็นจุดที่มีค่ามากกว่าพื้นที่รอบข้างซึ่งเรียกจุด L ว่าจุดสุดสัมพัทธ์



รูปที่ 2.4 ตัวอย่างปัญหาหาค่าเหมาะที่สุดที่มีหลายจุดที่ดีที่สุด

2.1.3 ตัวอย่างปัญหาการหาจุดที่ดีที่สุด

ปัญหาการหาจุดที่ดีที่สุดสามารถพบได้ในปัญหาทางธุรกิจ [76] เช่น ปัญหาการเลือกโครงการ (Project Selection Problem) ปัญหาจัดสรรเงินลงทุน (Capital Budgeting Problem) ปัญหาจัดการคลังสินค้า (Cutting Stock Problem) ปัญหาจัดการโหลดสินค้า (Cargo Loading Problem) ปัญหาการวางแผนการผลิต (Production Planning Problem) ปัญหาการจัดตารางเวลา (Scheduling Problem) ปัญหาจัดการทรัพยากรการประมวลผล (Distributed Processor Problem) ปัญหาการจัดสรรฐานข้อมูล (Database Allocation Problem) และปัญหาถุงเป้ (Knapsack Problem หรือ KP) เป็นต้น นอกจากนี้ยังพบในปัญหาทางการขนส่ง [14] เช่น

ปัญหาจัดการเส้นทางการขนส่ง (Logistics and Transportation Problem) ปัญหาการหาเส้นทางที่สั้นที่สุด (Shortest Path Problem) ปัญหาการไหลสูงสุด (Maximum Flow Problem) ปัญหาการมอบหมายหน้าที่ (Generalized Assignment Problem) ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem หรือ TSP) และปัญหาการจัดการเส้นทางรถยนต์ (Vehicle Routing Problem) เป็นต้น นอกจากนี้ยังมีปัญหาทางด้านอื่นๆ อีกมากมายที่เป็นปัญหาการหาจุดที่ดีที่สุดเช่น ปัญหาการผสมสร้างอาหาร (Blending Problem) ปัญหาการวางแผนการเพาะปลูก (Crop Planning Problem) ปัญหาการออกแบบเครื่องบินให้มีน้ำหนักต่ำสุดและมีความแข็งแรงสูงสุด ปัญหาการออกแบบโครงสร้างของอาคารให้มีค่าใช้จ่ายที่ต่ำที่สุดและแข็งแรงที่สุด ปัญหาที่ต้องการหาเส้นทางโคจรของดาวเทียมให้ได้ระยะที่เหมาะสมที่สุด เป็นต้น

2.2 การหาจุดที่ดีที่สุด

2.2.1 นิยามของการหาจุดที่ดีที่สุด

การหาจุดที่ดีที่สุดหรือการหาค่าที่ดีที่สุด (Optimization) คือการหาค่าสูงสุดหรือค่าต่ำสุดของปัญหาการหาจุดที่ดีที่สุด [64] หรือการหาค่าตัวแปรต้นที่ทำให้ผลลัพธ์จากฟังก์ชันเป้าหมายมีค่ามากที่สุดหรือค่าน้อยที่สุด [65]

2.2.2 ประเภทของขั้นตอนวิธีการหาจุดที่ดีที่สุด

2.2.2.1 ประเภทขั้นตอนวิธีพื้นฐานเส้นโคจรและประชากร

ขั้นตอนวิธีพื้นฐานเส้นโคจรหรือขั้นตอนวิธีพื้นฐานเส้นโคจร (Trajectory-Based Algorithm) [79] คือขั้นตอนวิธีที่ใช้ตัวแทน (Agent) เพียงตัวเดียวเท่านั้นสำหรับการแก้ปัญหาการหาจุดที่ดีที่สุดตัวอย่างเช่น ขั้นตอนวิธีกลวิธีการค้ำอย่างไต่เขา (Hill-Climbing Algorithm) ขั้นตอนวิธีการจำลองการอบเหนียว (Simulated Annealing Algorithm หรือ SA) [75] เป็นต้น

ขั้นตอนวิธีพื้นฐานประชากรหรือขั้นตอนวิธีพื้นฐานประชากร (Population-based Algorithm) [79] คือขั้นตอนวิธีที่ใช้ตัวแทนหลายตัวสำหรับแก้ปัญหาการหาจุดที่ดีที่สุดตัวอย่างเช่น GA [5, 6], PSO [1, 2] เป็นต้น

2.2.2.2 ประเภทขั้นตอนวิธีการค้นหาอย่างแน่นอนและไม่แน่นอน

ขั้นตอนวิธีการค้นหาอย่างแน่นอนหรือขั้นตอนวิธีการค้นหาอย่างแน่นอน (Deterministic Algorithm) [79] คือขั้นตอนวิธีที่ไม่มีขั้นตอนการสุ่มค่าในขั้นตอนการค้นหา ดังนั้นถ้าใช้ขั้นตอนวิธีเดียวกันและเริ่มต้นในตำแหน่งเดียวกัน จะได้ผลลัพธ์การค้นหาเหมือนเดิมทุกครั้ง ตัวอย่างขั้นตอนวิธีนี้คือ ขั้นตอนวิธีกลวิธีการค้ำอย่างไต่เขา เป็นต้น

ขั้นตอนวิธีการค้นหาอย่างไม่แน่นอนหรือขั้นตอนวิธีการค้นหาอย่างไม่แน่นอน (Stochastic Algorithm) [79] คือขั้นตอนวิธีที่มีขั้นตอนการสุ่มค่าอยู่ในบางขั้นตอนการค้นหา ดังนั้นผลลัพธ์การค้นหาจะมีโอกาสได้ผลลัพธ์ที่แตกต่างกันถึงแม้ว่าจะเริ่มต้นด้วยจุดเดียวกัน และใช้ขั้นตอนวิธีเดียวกัน ตัวอย่างขั้นตอนวิธีนี้คือ GA, PSO, SA เป็นต้น

2.2.2.3 ประเภทขั้นตอนวิธีใช้และไม่ใช้ผลการค้นหาเดิม

ขั้นตอนวิธีที่ไม่ใช้ผลการค้นหาเดิม (Non History-based Algorithm) [79] คือขั้นตอนวิธีที่มีการกำหนดการใช้หน่วยความจำคงที่ โดยจะใช้หน่วยความจำบันทึกเฉพาะสถานะที่ดีที่สุดในปัจจุบันเท่านั้น สำหรับผลการค้นหาเดิมทั้งหมดจะถูกลบทิ้ง ตัวอย่างขั้นตอนวิธีนี้คือ GA, PSO เป็นต้น

ขั้นตอนวิธีที่ใช้ผลการค้นหาเดิม (History-based Algorithm) [79] คือขั้นตอนวิธีที่มีการใช้หน่วยความจำที่ไม่คงที่ มีการบันทึกผลการค้นหาเดิมเพื่อนำมาใช้ในการพิจารณาการค้นหาในครั้งต่อไป ตัวอย่างขั้นตอนวิธีนี้คือการค้นหาแบบข้อห้าม (TABU search) [80] เป็นต้น

2.2.3 พฤติกรรมการค้นหาของวิธีการหาจุดที่ดีที่สุด

วิธีการหาค่าการหาจุดที่ดีที่สุดสามารถแบ่งพฤติกรรมของการค้นหาได้ 2 แบบ คือการค้นหาแบบหนาแน่น (Intensification Search) [81] คือการค้นหาโดยที่ตัวแทนจะค้นหาคำตอบในบริเวณใดบริเวณหนึ่งอย่างละเอียดและการค้นหาจะค้นหาในบริเวณขอบเขตที่กำหนด ตัวแทนจะมีลักษณะการลู่เข้ามารวม ณ จุดใดจุดหนึ่ง สำหรับการค้นหาอีกแบบ คือการค้นหาแบบกระจาย (Diversification Search) [81] คือการค้นหาโดยที่ตัวแทนจะกระจายตัวไปค้นหาคำตอบในบริเวณกว้างและการค้นหาจะค้นหาอยู่นอกขอบเขตที่กำหนด ตัวแทนจะมีลักษณะกระจายตัวออกจากจุดใดจุดหนึ่ง

2.3 ปัญหาการติดในจุดสุดสัมพัทธ์

2.3.1 จุดสุดสัมพัทธ์และจุดสุดสัมบูรณ์

ค่าสูงสุดสัมพัทธ์ (Local Maximum) คือค่าของฟังก์ชันที่นิยามจากจุดวิกฤต (x_c, y_c) ในช่วง (a, b) บนโดเมนของฟังก์ชันที่ $f(x) \leq f(x_c)$ เรียกจุดวิกฤตนั้นว่า จุดสูงสุดสัมพัทธ์ โดยตำแหน่งของจุดนี้ได้แสดงไว้บนกราฟ 2.5

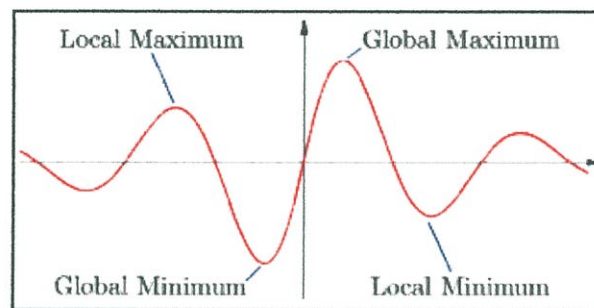
ค่าต่ำสุดสัมพัทธ์ (Local Minimum) คือค่าของฟังก์ชันที่นิยามจากจุดวิกฤต (x_c, y_c) ในช่วง (a, b) บนโดเมนของฟังก์ชันที่ $f(x) \geq f(x_c)$ เรียกจุดวิกฤตนั้นว่า จุดต่ำสุดสัมพัทธ์ โดยตำแหน่งของจุดนี้ได้แสดงไว้บนกราฟ 2.5

ค่าสูงสุดสัมบูรณ์ (Global Maximum) คือค่าของฟังก์ชันที่นิยามจากจุดวิกฤต (x_c, y_c) บนช่วง $[a, b]$ ของโดเมนของฟังก์ชันที่ $f(x) \leq f(x_c)$ เรียกจุดวิกฤตนั้นว่า จุดสูงสุดสัมบูรณ์ โดยตำแหน่งของจุดนี้ได้แสดงไว้บนกราฟ 2.5

ค่าต่ำสุดสัมบูรณ์ (Global Maximum) คือค่าของฟังก์ชันที่นิยามจากจุดวิกฤต (x_c, y_c) บนช่วง $[a, b]$ ของโดเมนของฟังก์ชันที่ $f(x) \geq f(x_c)$ เรียกจุดวิกฤตนั้นว่า จุดต่ำสุดสัมบูรณ์ โดยตำแหน่งของจุดนี้ได้แสดงไว้บนกราฟ 2.5

จุดสุดสัมพัทธ์ (Local Optimum) [72] คือจุดบนค่าของฟังก์ชัน ซึ่งเป็นค่าสูงสุดหรือต่ำสุด (จุดสูงสุดสัมพัทธ์หรือจุดต่ำสุดสัมพัทธ์) ในบริเวณใดบริเวณหนึ่งของโดเมน สำหรับค่านี้อาจจะมีได้ไม่จำกัดต่อหนึ่งฟังก์ชัน

จุดสุดสัมบูรณ์ (Global Optimum) [73] คือจุดบนค่าของฟังก์ชัน ซึ่งเป็นค่าสูงสุดหรือต่ำสุด (จุดสูงสุดสัมบูรณ์หรือจุดต่ำสุดสัมบูรณ์) ตลอดโดเมนของฟังก์ชัน สำหรับค่านี้จะมีได้ไม่เกินหนึ่งค่าต่อหนึ่งฟังก์ชัน

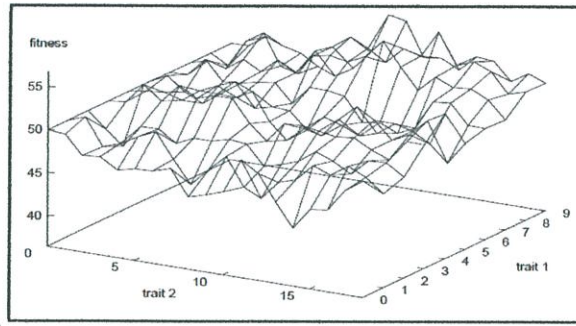


รูปที่ 2.5 จุดสุดสัมพัทธ์และจุดสุดสัมบูรณ์ [77]

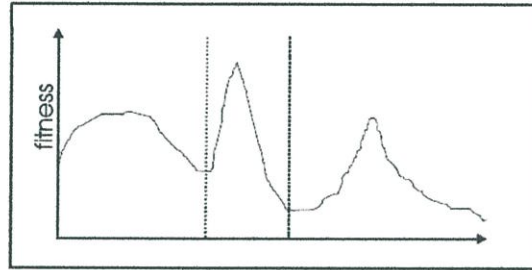
2.3.2 นิยามของปัญหาการติดในจุดสุดสัมพัทธ์

ปัญหาการติดในจุดสุดสัมพัทธ์ (Local Optimum Problem) คือการดำเนินการหาค่าการหาจุดที่ดีที่สุดนั้นไม่สามารถพบจุดสุดสัมบูรณ์ อันเนื่องมาจากตัวแทนนั้นได้ติดอยู่ในบริเวณจุดสุดสัมพัทธ์และตัวแทนเหล่านี้ไม่ออกไปค้นหาในบริเวณอื่นอีก นอกจากบริเวณรอบจุดสุดสัมพัทธ์นั้น จึงไม่สามารถที่จะค้นพบจุดสุดสัมบูรณ์ ไม่ว่าจะดำเนินการค้นหาต่อไปอีกยาวนานเท่าใด ก็จะไม่มีโอกาสพบคำตอบที่ดีกว่าจุดสุดสัมพัทธ์ที่ติดอยู่นั้น โดยปัญหาการติดในจุดสุดสัมพัทธ์จะเกิดขึ้นกับปัญหามัลติโมดอล [71] เท่านั้น ตัวอย่างปัญหามัลติโมดอลคือรูป 2.6

จากรูป 2.6 จะมียอดเขาจำนวนมาก การหาจุดที่ดีที่สุดนั้นจะมีโอกาสที่จะพบยอดเขาที่ไม่ใช่ยอดเขาที่สูงสุด ซึ่งวิธีการค้นหาคำตอบแบบเฉพาะที่ (Local Search) จะค้นหายอดเขาที่สูงสุดได้ยากมาก โดยพิจารณาจากรูป 2.7 [71] ซึ่งเป็นรูปที่ตัดพื้นที่บางส่วนมาจากรูป 2.6 แล้วแสดงในมุมมอง 2 มิติ จากรูปจะพบว่าในช่วงระหว่างเส้นประ ถ้าวิธีการค้นหาคำตอบแบบเฉพาะที่เริ่มต้นค้นหาในช่วงระหว่างนี้ก็มีโอกาสพบจุดสุดสัมบูรณ์ แต่ถ้าเริ่มต้นค้นหาในช่วงบริเวณอื่นจะไม่มีโอกาสพบจุดสุดสัมบูรณ์ [71]



รูปที่ 2.6 ตัวอย่างของปัญหามัลติโมดอลในมุมมองสามมิติ [71]

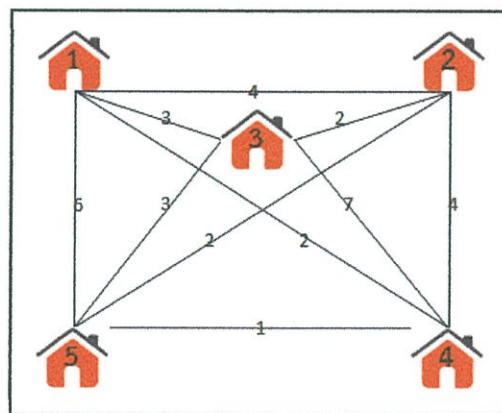


รูปที่ 2.7 ตัวอย่างของปัญหามัลติโมดอลในมุมมองสองมิติ [71]

2.3.3 ตัวอย่างของปัญหาการติดในจุดสุดสัมพัทธ์

การแก้ปัญหาค้นหาจุดที่ดีที่สุดโดยทั่วไปนั้น สามารถเกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้หลากหลายกรณี ตัวอย่างเช่น ปัญหาการขนส่งสินค้าให้ได้ระยะทางสั้นที่สุด โดยรูปแบบเส้นทางขนส่งสินค้าเป็นไปตามรูป 2.8 จากรูปนี้มีเงื่อนไขว่าให้ขนส่งสินค้าไปครบทุกสถานที่ โดยมีระยะทางสั้นที่สุด ซึ่งระยะทางคือค่าของตัวเลขระหว่างเส้นทางในรูป

สำหรับขั้นตอนวิธีที่ใช้นั้นจะสามารถทำการสลับเปลี่ยนสถานที่ได้ที่ละหนึ่งคู่ที่อยู่ข้างเคียงกันเท่านั้น จากรูป 2.8 คำตอบจากปัญหานี้ คือลำดับสถานที่ดังนี้ 1, 4, 5, 2, 3 ซึ่งจะได้ระยะรวมทั้งหมด คือ 7 กิโลเมตร



รูปที่ 2.8 ตัวอย่างกรณีศึกษาปัญหาการติดในจุดสุดสัมพัทธ์

สมมติว่าขั้นตอนวิธีที่ใช้กันสามารถสลับเปลี่ยนลำดับของสถานที่จนออกมาได้ผลลัพธ์ดังนี้ 1, 3, 5, 4, 2 ซึ่งจะได้ระยะรวมทั้งหมด คือ 11 กิโลเมตร ถ้าขั้นตอนวิธีที่ใช้กันสามารถสลับเปลี่ยนได้ทีละหนึ่งคู่ที่อยู่ข้างเคียงกัน ผลลัพธ์ที่ได้จะเป็นดังนี้ กรณีที่หนึ่งสลับที่คู่แรก 3, 1, 5, 4, 2 มีระยะรวมทั้งหมด คือ 14 กิโลเมตร กรณีที่สองสลับที่คู่สอง 1, 5, 3, 4, 2 มีระยะรวมทั้งหมด คือ 20 กิโลเมตร กรณีที่สามสลับที่คู่สาม 1, 3, 4, 5, 2 มีระยะรวมทั้งหมด คือ 13 กิโลเมตร กรณีที่สี่สลับที่คู่สี่ 1, 3, 5, 2, 4 มีระยะรวมทั้งหมด คือ 12 กิโลเมตร

จะสังเกตเห็นว่าไม่ว่าจะสลับอย่างไรก็ตามคำตอบที่ได้ก็แย่กว่าเดิมเสมอ ดังนั้นคำตอบที่ได้จากค้นหาครั้งนี้ คือ 11 แต่ที่จริงแล้วคำตอบของปัญหา คือ 7 ซึ่งสถานการณ์นี้ เรียกว่า ปัญหาการติดในจุดสุดสัมพัทธ์ โดยเมื่อเกิดปัญหานี้ขึ้นแล้วจะไม่สามารถหาคำตอบที่ดีขึ้นกว่าเดิมได้

2.3.4 สาเหตุของปัญหาการติดในจุดสุดสัมพัทธ์

จากกรณีศึกษาพบว่าสาเหตุที่ทำให้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ คือ ขั้นตอนวิธีที่นำมาใช้และความซับซ้อนของปัญหา ถ้าขั้นตอนวิธีที่นำมาใช้นั้นมีประสิทธิภาพเพียงพอที่จะแก้ปัญหา ก็จะสามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้ เช่น ในกรณีศึกษาถ้าขั้นตอนวิธีที่ใช้กันสามารถสลับลำดับได้สองคู่ที่อยู่ข้างเคียงกัน จากกรณีผลลัพธ์ที่ออกมาคือ 1, 3, 5, 4, 2 ซึ่งจะได้ระยะรวมทั้งหมด คือ 11 กิโลเมตร ถ้าใช้ขั้นตอนวิธีนี้ เมื่อสลับที่คู่แรก ผลลัพธ์ที่ได้ คือ 1, 3, 5, 2, 4 และเมื่อสลับที่คู่ที่สอง ผลลัพธ์ที่ได้ คือ 1, 3, 2, 5, 4 ซึ่งระยะทางรวมทั้งหมด คือ 8 กิโลเมตร การนำขั้นตอนวิธีนี้มาใช้ส่งผลให้เกิดการหลุดจากการติดในจุดสุดสัมพัทธ์ นอกจากนี้ถ้าปัญหาที่แก้ไอนั้นมีความซับซ้อนน้อยโอกาสที่จะเกิดปัญหาการติดในจุดสุดสัมพัทธ์ก็จะน้อย เช่น ในกรณีศึกษาถ้าลดสถานที่เหลือแค่ 3 สถานที่ แต่ใช้ขั้นตอนวิธีการสลับที่สถานที่เพียงหนึ่งคู่ที่อยู่ข้างเคียงกัน สามารถค้นพบจุดสุดสัมบูรณ์ได้อย่างแน่นอน เพราะขั้นตอนวิธีนี้สามารถค้นหาคำตอบที่เป็นไปได้ทั้งหมดของปัญหา

ในทางปฏิบัติปัญหาการหาจุดที่ดีที่สุดโดยส่วนมากเป็นปัญหามัลติโมดอล [71] จากที่ได้กล่าวไปแล้ว ปัญหาประเภทนี้สามารถทำให้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ โดยสาเหตุของการเกิดปัญหาการติดในจุดสุดสัมพัทธ์ คือความซับซ้อนของปัญหาการหาจุดที่ดีที่สุดและขั้นตอนวิธีที่นำมาแก้ปัญหา ถ้าต้องการแก้ปัญหาการติดในจุดสุดสัมพัทธ์ สิ่งที่ไม่สามารถทำได้คือลดความซับซ้อนของปัญหาการหาจุดที่ดีที่สุด เพราะจะเป็นการเปลี่ยนไปแก้ปัญหาการหาจุดที่ดีที่สุดอื่น ดังนั้นการแก้ปัญหาการหาจุดที่ดีที่สุด จึงจำเป็นต้องเพิ่มประสิทธิภาพของขั้นตอนวิธีที่นำมาแก้ปัญหานั้น

2.4 วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค

2.4.1 นิยามของ PSO

วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO) เป็นขั้นตอนวิธีที่นำมาใช้กับการแก้ปัญหาการหาจุดที่ดีที่สุด โดย PSO เป็นขั้นตอนวิธีที่มีสมบัติเป็นขั้นตอนวิธีพื้นฐานประชากร ขั้นตอนวิธีการค้นหาอย่างไม่แน่นอน และขั้นตอนวิธีที่ไม่ใช้ผลการค้นหาเดิม โดยที่ PSO เป็นหนึ่งในกลุ่มของขั้นตอนวิธีวิวัฒนาการ (Evolutionary Algorithm หรือ

EA) ซึ่งได้ถูกนำเสนอครั้งแรก โดย เจมส์ เคนเนดี และ รัสเซล อีเบอฮาท [1, 2] ในปี 1995 โดยมีแรงบันดาลใจจากพฤติกรรมการบินของฝูงสัตว์ เช่น ฝูงนก ฝูงปลา เป็นต้น และการสื่อสารแลกเปลี่ยนข้อมูลระหว่างพวกฝูงสัตว์เหล่านั้นในขณะที่ออกหาอาหาร โดยฝูงสัตว์เหล่านี้ จะทำการเคลื่อนที่เข้าหาแหล่งอาหาร แล้วส่งสัญญาณสื่อสารกันภายในฝูง เพื่อให้ทราบถึงตำแหน่งที่มีอาหารอยู่ และทำให้สัตว์ภายในฝูงตัวอื่นทำการเคลื่อนที่ไปยังแหล่งอาหารเหล่านั้น ซึ่งแนวคิดนี้ตัวแทนของสัตว์ที่ใช้ในการค้นหาจะถูกเรียกว่า อนุภาค (Particles หรือ X) โดยอนุภาคนั้นจะมีจำนวนมากและเคลื่อนที่บนพื้นที่ที่ต้องการค้นหาคำตอบหรือปริภูมิของคำตอบ (Solution Space) ซึ่งเป็นตัวแทนของแหล่งอาหาร โดยอนุภาคจะต้องค้นหาคำตอบที่ดีที่สุด (Optimum) ซึ่งเป็นตัวแทนของแหล่งอาหารที่มากที่สุด

PSO สามารถประยุกต์ใช้กับทุกปัญหาที่คำตอบสามารถแทนด้วยจุดบนปริภูมิขนาดหลายมิติ โดยอนุภาคจะถูกวางไว้ในปริภูมิของคำตอบพร้อมกับความเร็วต้นค่าหนึ่งและช่องทางในการติดต่อกับอนุภาคอื่นๆ [1, 2] อนุภาคจะเคลื่อนที่ไปเรื่อยๆ บนปริภูมิของคำตอบ โดยอนุภาคที่เข้าใกล้คำตอบมากที่สุดจะดึงดูดให้อนุภาคตัวอื่นเปลี่ยนแปลงเส้นทางการเคลื่อนที่ให้ลู่อเข้ามาหาอนุภาคที่เข้าใกล้คำตอบมากที่สุด

2.4.2 นิยามที่ใช้กับ PSO

2.4.2.1 อนุภาค

อนุภาค (Particles) คือตัวแทนคำตอบของปัญหาที่ต้องการค้นหา โดยการกำหนดตำแหน่งของอนุภาคจะอยู่ในขอบเขตของปริภูมิของคำตอบ โดยแต่ละอนุภาคจะมีองค์ประกอบของตัวเอง นั่นคือความเร็วปัจจุบันของอนุภาค (V) ตำแหน่งปัจจุบันของอนุภาค (X) และตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัว (PBEST)

2.4.2.2 ประชากร

ประชากร (Population) คือกลุ่มของอนุภาคที่ใช้เคลื่อนที่เพื่อหาคำตอบในปริภูมิของคำตอบ โดยจะมีการกำหนดจำนวนประชากร (N_{pop}) เอาไว้ล่วงหน้า และจะมีจำนวนคงที่ตลอดการค้นหา

2.4.2.3 รอบการค้นหาคำตอบ

รอบการค้นหาคำตอบ ($Iter$) คือจำนวนรอบที่ใช้ในการค้นหาคำตอบ เนื่องจาก PSO จะค้นหาคำตอบโดยทำงานแบบวนซ้ำไปเรื่อยๆ รอบของการค้นหาคำตอบจึงเพิ่มขึ้นไปเรื่อยๆ จนกว่าจะถึงรอบการค้นหาคำตอบสูงสุด ($Iter_{max}$) ที่ได้กำหนดไว้ให้หยุดการทำงาน

2.4.2.4 ความเร็วของอนุภาค

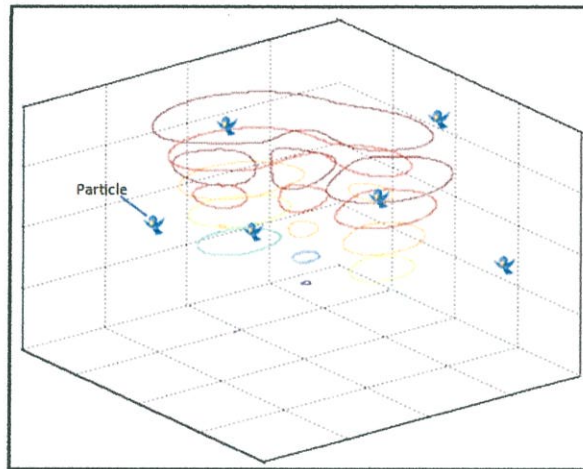
ความเร็วของอนุภาค (Velocity หรือ V) คือขนาดและทิศทางที่อนุภาคจะเคลื่อนที่ไปจากตำแหน่งเดิมสู่ตำแหน่งใหม่

2.4.2.5 ความเร็วสูงสุด

ความเร็วสูงสุด (V_{max}) คือค่าสูงสุดที่เป็นไปได้ของความเร็วของอนุภาคเพื่อจำกัดระยะทางที่มากที่สุดที่อนุภาคสามารถเคลื่อนที่ได้ในหนึ่งรอบการค้นหาคำตอบ

2.4.2.6 ตำแหน่งของอนุภาค

ตำแหน่งของอนุภาค (Position หรือ X) คือตำแหน่งของอนุภาคในปริภูมิคำตอบ เช่น รูป 2.9 แสดงตัวอย่างของตำแหน่งอนุภาคในปริภูมิคำตอบที่มีลักษณะเป็นสามมิติ



รูปที่ 2.9 ตำแหน่งอนุภาคในแกนสามมิติ [77]

2.4.2.7 ตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัว

ตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัวหรือค่าความเหมาะสมที่ดีที่สุดของอนุภาคแต่ละตัว (PBEST หรือ Personal Best Position) คือตำแหน่งของอนุภาคที่มีค่าความเหมาะสมที่ดีที่สุดเท่าที่อนุภาคตัวนั้นๆ เคยทำการค้นหาคำตอบมาตั้งแต่เริ่มการทำงาน

2.4.2.8 ตำแหน่งที่ดีที่สุดของอนุภาคทั้งหมด

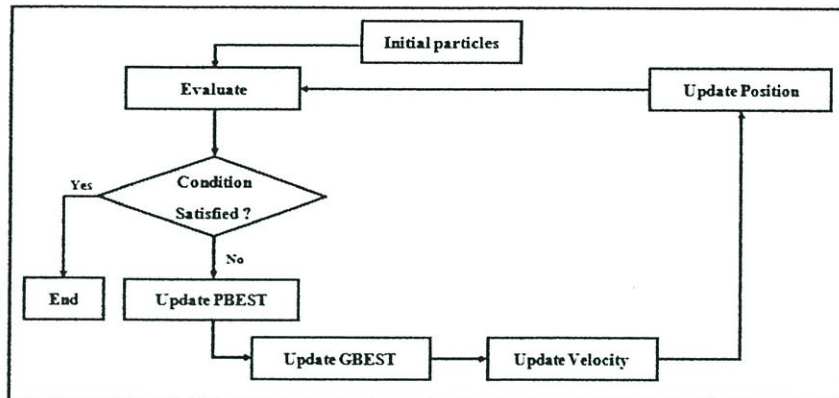
ตำแหน่งที่ดีที่สุดของอนุภาคทั้งหมดหรือค่าความเหมาะสมที่ดีที่สุดของอนุภาคทั้งหมด คือ (GBEST หรือ Global Best Position) ตำแหน่งของอนุภาคที่มีค่าความเหมาะสมดีที่สุดเท่าที่อนุภาคทั้งหมดเคยทำการค้นหาตั้งแต่เริ่มการทำงาน

2.4.2.9 ฟังก์ชันคำนวณค่าความเหมาะสมหรือฟังก์ชันเป้าหมาย

ฟังก์ชันคำนวณค่าความเหมาะสม (Fitness Function) หรือ ฟังก์ชันเป้าหมาย (Objective Function) คือฟังก์ชันที่ใช้คำนวณค่าความเหมาะสม (Fitness) ของแต่ละอนุภาค เพื่อใช้เปรียบเทียบและหาอนุภาคที่ดีที่สุด ซึ่งฟังก์ชันนี้จะแตกต่างกันขึ้นอยู่กับปัญหาที่ต้องการค้นหา

2.4.3 หลักการทำงานของ PSO

ขั้นตอน PSO มีส่วนใกล้เคียงกับ GA หลายประการ เช่น การสุ่มค่าเริ่มต้นและการปรับปรุงคำตอบให้ดีขึ้นในทุกรอบการค้นหาคำตอบ ส่วนสิ่งที่แตกต่างคือวิธีการปรับปรุงคำตอบในแต่ละรอบจะใช้วิธีการเคลื่อนที่เข้าสู่ค่าที่ดีขึ้นเรื่อยๆ แต่ GA ใช้วิธีการวิวัฒนาการ เช่น การผสมพันธุ์หรือ การกลายพันธุ์ โดยหลักการทำงานของ PSO มีขั้นตอนการทำงานดังแสดงในรูป 2.10



รูปที่ 2.10 แผนภาพการทำงานของ PSO

2.4.3.1 กระบวนการกำหนดค่าเริ่มต้น

กระบวนการกำหนดค่าเริ่มต้น (Initial Particles) เป็นกระบวนการที่ทำงานเป็นขั้นตอนแรกและทำงานเพียงครั้งเดียวเท่านั้น โดยสุ่มตำแหน่งและความเร็วเริ่มต้นของอนุภาคทั้งหมดลงบนปริภูมิขนาดหลายมิติตามจำนวนประชากรที่ได้กำหนดเอาไว้ โดยความเร็วที่สุ่มขึ้นมาจะต้องไม่เกินค่าของความเร็วสูงสุดที่กำหนดเอาไว้ หากความเร็วที่สุ่มขึ้นมา มีค่ามากกว่าค่าความเร็วสูงสุดให้เปลี่ยนมาใช้ค่าความเร็วสูงสุดแทน โดยจำนวนประชากรถ้าหากมีจำนวนมากการค้นหาคำตอบจะทำได้กว้างและรวดเร็ว แต่ต้องใช้เวลาในแต่ละรอบการค้นหาคำตอบเพิ่มขึ้น ดังนั้นแต่ละปัญหาจึงไม่จำเป็นต้องกำหนดจำนวนอนุภาคให้เท่ากันเสมอไป โดยจำนวนอนุภาคที่สร้างขึ้นนี้ จะไม่มีการเพิ่มหรือลดจำนวนตลอดการทำงาน

2.4.3.2 กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค

กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค (Evaluate) เป็นกระบวนการคำนวณค่าความเหมาะสม ณ ตำแหน่งปัจจุบันของแต่ละอนุภาค โดยค่าความเหมาะสมเหล่านี้จะถูกใช้ในการปรับค่า PBEST และ GBEST นอกจากนี้จะทำการตรวจสอบคำตอบว่าค่าความเหมาะสมของอนุภาคได้เคลื่อนที่ไปพบคำตอบหรือไม่ ถ้าพบคำตอบหรือจำนวนรอบการค้นหาคำตอบเท่ากับจำนวนรอบการค้นหาคำตอบสูงสุดที่กำหนดเอาไว้ก็จะจบการค้นหานั้นก็จะดำเนินขั้นตอนต่อไป

2.4.3.3 กระบวนการเปลี่ยนแปลง PBEST

กระบวนการเปลี่ยนแปลง PBEST (Update PBEST) เป็นการเปลี่ยนแปลง PBEST โดยใช้ฟังก์ชันคำนวณค่าความเหมาะสมคำนวณหาค่าความเหมาะสมในตำแหน่งที่อนุภาคอยู่ปัจจุบัน หากค่าความเหมาะสมที่คำนวณได้นั้นมีค่าดีกว่า PBEST ก็ให้ PBEST เปลี่ยนมาใช้ค่าใหม่แทน สำหรับ PBEST นี้จะใช้ในการคำนวณความเร็วและทิศทางใหม่ให้กับอนุภาค เพื่อให้อนุภาคเหล่านั้นเคลื่อนที่ไปยังตำแหน่งใหม่

2.4.3.4 กระบวนการเปลี่ยนแปลง GBEST

กระบวนการเปลี่ยนแปลง GBEST (Update GBEST) เป็นการเปลี่ยนแปลง GBEST โดยใช้ฟังก์ชันคำนวณค่าความเหมาะสม ซึ่งจะทำการคำนวณหาค่าความเหมาะสมในตำแหน่งที่อนุภาคอยู่ปัจจุบัน หากค่าความเหมาะสมที่คำนวณได้นั้นมีค่าดีกว่า GBEST ก็ให้ GBEST เปลี่ยนมาใช้ค่าใหม่แทน ซึ่ง GBEST จะทำการเปรียบเทียบกับค่าความเหมาะสมของอนุภาคทุกตัว สำหรับ GBEST นี้จะใช้ในการคำนวณความเร็วและทิศทางใหม่ให้กับอนุภาค เพื่อใช้เคลื่อนที่อนุภาคเหล่านั้นไปยังตำแหน่งใหม่ GBEST คือ ค่าตอบของการค้นหาด้วย PSO

2.4.3.5 กระบวนการคำนวณความเร็ว

กระบวนการคำนวณความเร็วหรือกระบวนการเปลี่ยนแปลงความเร็ว (Calculate Velocity หรือ Update Velocity) เป็นกระบวนการที่ทำการคำนวณความเร็วของแต่ละอนุภาค เพื่อใช้ในการเคลื่อนที่อนุภาคแต่ละตัวในแต่ละรอบ ซึ่งความเร็วนี้จะถูกเปลี่ยนแปลง โดยใช้ข้อมูลของความเร็วเดิมของอนุภาคแต่ละตัว PBEST และ GBEST โดยมีสมการในการคำนวณดังนี้

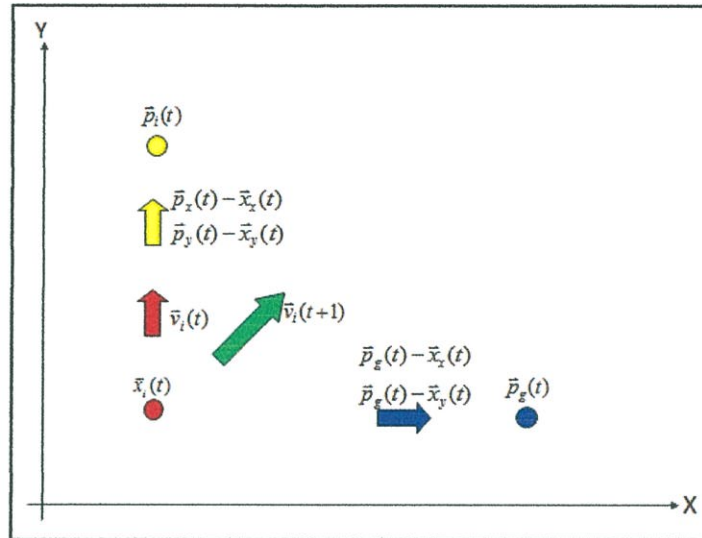
$$\vec{v}_i(t+1) = \omega \vec{v}_i(t) + c_1 \phi_1 (\vec{p}_i(t) - \vec{x}_i(t)) + c_2 \phi_2 (\vec{p}_g(t) - \vec{x}_i(t)) \quad (2.2)$$

โดยกำหนดให้

$\vec{v}_i(t+1)$	คือ ความเร็วในรอบปัจจุบันเป็นเวกเตอร์ที่มีขนาดและทิศทาง
$\vec{v}_i(t)$	คือ ความเร็วในรอบก่อนหน้าเป็นเวกเตอร์ที่มีขนาดและทิศทาง
c_1, c_2	คือ ค่าคงที่ทำหน้าที่เป็นสัมประสิทธิ์
ϕ_1, ϕ_2	คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วง $[0, 1]$
ω	คือ ค่าความเฉื่อยทำหน้าที่เป็นสัมประสิทธิ์
$\vec{p}_i(t)$	คือ ตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัว (PBEST)
$\vec{p}_g(t)$	คือ ตำแหน่งที่ดีที่สุดของอนุภาคทั้งหมด (GBEST)
$\vec{x}_i(t)$	คือ ตำแหน่งในปัจจุบันของอนุภาค i (X)

โดยค่าของ \vec{v}_i จะต้องไม่เกินค่าของความเร็วสูงสุด การเปลี่ยนความเร็วในลักษณะนี้จะทำให้อนุภาค i สามารถค้นหาค่าตอบที่อยู่รอบๆ PBEST และ GBEST ได้ดังรูปที่ 2.11 แสดงตัวอย่างการคำนวณทิศทางความเร็วของอนุภาคในแกนสองมิติ

จากสมการที่ 2.2 พารามิเตอร์ ω คือ ความเฉื่อยซึ่งเป็นตัวแปรที่มีอิทธิพลควบคุมความเร็วในรอบก่อนหน้าที่ส่งผลต่อการเคลื่อนที่ของอนุภาคในรอบปัจจุบัน c_1 คือ ค่าคงที่ซึ่งเป็นตัวกำหนดแรงดึงดูดจากอิทธิพลของ PBEST c_2 คือ ค่าคงที่ซึ่งเป็นตัวกำหนดแรงดึงดูดจากอิทธิพลของ GBEST ซึ่งผลจากสมการนี้ ทำให้อนุภาคเคลื่อนที่ไปยังตำแหน่งที่ดีที่สุด



รูปที่ 2.11 ตัวอย่างทิศทางการเคลื่อนที่ของอนุภาคในแกนสองมิติ [77]

รูปที่ 2.11 กำหนดให้ทิศทางของความเร็วในรอบก่อนหน้า $\vec{v}_i(t)$ มีทิศทางเป็นเวกเตอร์ลูกศรสีแดง สำหรับการคำนวณทิศทางของ PBEST (\vec{p}_i) ให้นำตำแหน่งของ PBEST ลบกับตำแหน่งของอนุภาค ทำให้เกิดเวกเตอร์ลูกศรสีเหลือง และการคำนวณทิศทางของ GBEST (\vec{p}_g) ให้นำตำแหน่งของ GBEST ลบกับตำแหน่งของอนุภาค ทำให้เกิดเวกเตอร์ลูกศรสีน้ำเงิน และเมื่อนำเวกเตอร์ทั้งสามมารวมกันจะได้ทิศทางลัพท์ดังเวกเตอร์ลูกศรสีเขียว คือความเร็วในรอบปัจจุบัน $\vec{v}_i(t+1)$ ซึ่งเป็นทิศทางการเคลื่อนที่ใหม่ของอนุภาค

2.4.3.6 กระบวนการเปลี่ยนตำแหน่ง

กระบวนการเปลี่ยนตำแหน่ง (Update Position) คือการปรับปรุงตำแหน่งของอนุภาคใดๆ โดยอนุภาคนั้นๆ จะเคลื่อนที่จากตำแหน่งเดิมไปยังตำแหน่งใหม่ ด้วยระยะที่คำนวณได้จากค่าความเร็วของอนุภาคนั้นๆ โดยมีสมการดังนี้

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (2.3)$$

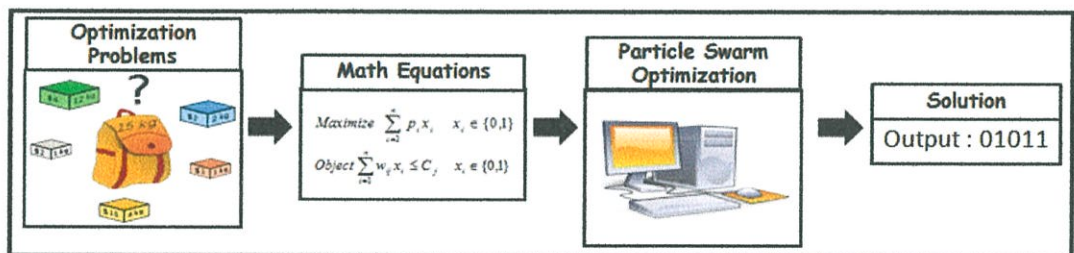
โดยกำหนดให้

- $\vec{x}_i(t+1)$ คือ ตำแหน่งใหม่ที่อนุภาคจะทำการเคลื่อนที่ไป
- $\vec{x}_i(t)$ คือ ตำแหน่งในปัจจุบันของอนุภาค i
- $\vec{v}_i(t+1)$ คือ ความเร็วใหม่ที่คำนวณได้

หลังจากกระบวนการเปลี่ยนตำแหน่ง ก็จะมีวนกลับไปทำกระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาคในรอบถัดไป โดยหลักการทำงานของ PSO ในช่วงแรกของการค้นหาอนุภาคจะอยู่ห่างกันมาก ทำให้ระยะห่างระหว่าง GBEST กับอนุภาคมีค่ามาก ส่งผลให้อนุภาคเคลื่อนที่เข้าหากันอย่างรวดเร็ว ทำให้ไม่ต้องสูญเสียเวลาค้นหาอย่างละเอียดในบริเวณที่ค่าความเหมาะสมที่มีค่าน้อย เมื่อการค้นหาดำเนินต่อไปอนุภาคจะเคลื่อนที่เข้าใกล้กันมากขึ้น ทำให้ผลต่างระหว่าง GBEST กับอนุภาคมีค่าน้อยลง ส่งผลให้อนุภาคเคลื่อนที่เข้าหากันช้าลง อนุภาคจะค้นหาบริเวณ GBEST อย่างละเอียด ซึ่งหลักการของ PSO นั้นได้ตั้งสมมติฐานว่า ณ บริเวณรอบ GBEST นั้น จะมีโอกาสมากที่สุดที่จะค้นพบค่าความเหมาะสมที่ดีขึ้นกว่าเดิมหรือค้นพบคำตอบที่ดีที่สุดภายในบริเวณนี้

2.5 ตัวอย่างการประยุกต์ใช้ PSO ในทางปฏิบัติ

ตัวอย่างการประยุกต์ใช้ PSO ในทางปฏิบัติ จากรูป 2.12 คือในขั้นตอนแรกก็จะทำการแปลงปัญหาการหาจุดที่ดีที่สุดให้อยู่ในรูปสมการคณิตศาสตร์ ถ้าสามารถแก้ไขสมการคณิตศาสตร์ได้ ก็เท่ากับแก้ปัญหาการหาจุดที่ดีที่สุดได้เช่นกัน หลังจากนั้นนำสมการคณิตศาสตร์มาค้นหาคำตอบด้วย PSO เมื่อค้นหาคำตอบเสร็จจะได้ผลลัพธ์ ซึ่งผลลัพธ์ที่ได้ก็คือคำตอบของปัญหาการหาจุดที่ดีที่สุด



รูปที่ 2.12 ตัวอย่างการนำ PSO ประยุกต์ใช้ในทางปฏิบัติ [84]

2.6 ข้อดีและข้อเสียของ PSO

2.6.1 ข้อดีของ PSO

PSO สามารถประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดที่มีปริภูมิของคำตอบขนาดใหญ่ [84] โดยใช้เวลาค้นหาไม่นาน ในกรณีปัญหาการหาจุดที่ดีที่สุดที่มีปริภูมิของคำตอบขนาดเล็ก เช่น ปัญหาถุงเป้ มีวัตถุทั้งหมด 5 วัตถุ ในกรณีนี้สามารถใช้วิธีค้นหาทุกคำตอบ (Brute force method) แก้ปัญหานี้ได้ เนื่องจากสามารถไล่คำตอบให้ครบ โดยมีคำตอบทั้งหมดคือ 32 คำตอบ จึงสามารถพบคำตอบที่ดีที่สุดอย่างแน่นอน ในทางกลับกัน ในกรณีปัญหาการหาจุดที่ดีที่สุดที่มีปริภูมิของคำตอบขนาดใหญ่ เช่น ปัญหาถุงเป้ มีวัตถุทั้งหมด 500 วัตถุ ในกรณีนี้มีคำตอบทั้งหมด คือ 2^{500} วิธีค้นหาทุกคำตอบต้องใช้เวลานานมากเพื่อไล่คำตอบให้ครบ วิธีค้นหาทุกคำตอบจึงไม่เหมาะสม ดังนั้นควรใช้วิธีอื่น เช่น PSO เพราะหลักการทำงานของ PSO จะไม่ทำการค้นหาคำตอบทั้งหมด แต่จะค้นหาเฉพาะบริเวณที่คิดว่าน่าจะมีคำตอบที่ดีอย่างละเอียดเท่านั้น จึงเป็นการลดขอบเขตการค้นหา และสามารถค้นหาได้อย่างมีประสิทธิภาพ โดยปกติแล้วในบริเวณที่พบคำตอบที่ดี จะมีโอกาส

สูงที่จะได้พบคำตอบที่ดีที่สุด อย่างไรก็ตาม PSO ไม่สามารถรับประกันว่าคำตอบที่ได้จากการค้นหาจะเป็นคำตอบที่ดีที่สุด แต่คำตอบที่ได้จากการค้นหาของ PSO จะมีค่าที่ดีในระดับหนึ่ง

ขั้นตอนวิธีที่แก้ปัญหาโดยเฉพาะเจาะจง (Problem-Specific Algorithm) จะมีประสิทธิภาพในการแก้ปัญหาดีกว่า PSO อย่างไรก็ตามขั้นตอนวิธีเหล่านี้ สามารถแก้ปัญหาแบบใดแบบหนึ่งได้เท่านั้น [71] เช่น ขั้นตอนวิธีลิน-เคอร์นิกแฮน (Lin-Kernighan Heuristics หรือ K-Opt Heuristic) [82, 83] สามารถแก้ปัญหา TSP ได้อย่างมีประสิทธิภาพ แต่ไม่สามารถประยุกต์ใช้กับปัญหา KP ในทางกลับกันขั้นตอนวิธีการดำเนินการซ่อมแซม (Repair Operator Algorithm) [94] สามารถแก้ปัญหา KP ได้อย่างมีประสิทธิภาพ แต่ไม่สามารถประยุกต์ใช้กับปัญหา TSP ได้นอกจากนั้นการสร้างขั้นตอนวิธีที่แก้ปัญหาโดยเฉพาะเจาะจงนั้นมีความยุ่งยาก เพราะต้องมีความรู้สำหรับการแก้ปัญหาค้นหาจุดที่ดีที่สุดนั้นโดยเฉพาะ จึงสามารถสร้างขั้นตอนวิธีมาแก้ปัญหาค้นหาจุดที่ดีที่สุดนั้นได้ ซึ่งแตกต่างจากขั้นตอนวิธีที่แก้ปัญหาโดยไม่เฉพาะเจาะจง หรือ PSO คือเมื่อพัฒนาโปรแกรมเสร็จสามารถนำไปประยุกต์ใช้กับปัญหารูปแบบอื่นได้ โดยการปรับเปลี่ยนพารามิเตอร์ (Parameters) เพียงบางส่วนเท่านั้น เช่น จำนวนรอบการค้นหา จำนวนประชากร และข้อจำกัดของปัญหา เป็นต้น เพื่อให้เหมาะสมกับการแก้ปัญหา [71] ด้วยเหตุผลนี้ PSO จึงนำสามารถประยุกต์ใช้กับปัญหาค้นหาจุดที่ดีที่สุดได้ง่าย [9, 10]

การคำนวณของ PSO มีประสิทธิภาพที่ดี [10] เนื่องจากที่ได้กล่าวไปแล้วว่า PSO จะไม่ค้นหาคำตอบทั้งหมด แต่จะค้นหาคำตอบเฉพาะบริเวณที่พบคำตอบที่ดีเท่านั้น ซึ่งในบริเวณนั้นจะมีโอกาสมากพบคำตอบที่ดีขึ้นได้มาก

พารามิเตอร์ของ PSO มีจำนวนน้อยที่ต้องการการกำหนดค่า [10] จากสมการที่ 2.2 และ 2.3 จำนวนพารามิเตอร์ของ PSO ที่ต้องการกำหนดค่า คือ c_1 , c_2 , ω สำหรับพารามิเตอร์อื่นที่ต้องกำหนดค่า คือ จำนวนประชากร และจำนวนรอบการค้นหาคำตอบ เนื่องจาก PSO มีจำนวนพารามิเตอร์น้อย จึงทำให้สามารถนำ PSO ไปประยุกต์ใช้ในทางปฏิบัติได้ง่าย

ขั้นตอนการทำงานของ PSO มีรูปแบบไม่ซับซ้อน จึงง่ายต่อการศึกษาและทำให้สามารถเขียนโปรแกรมพัฒนาได้ง่าย [10] จากที่ได้กล่าวไปแล้ว PSO มีหลักการที่ง่าย คือ อนุภาคจะพยายามเคลื่อนที่เข้าหา GBEST โดยขั้นตอนหลักที่เป็นตัวให้เกิดการเคลื่อนที่เช่นนี้ มีเพียงขั้นตอนของกระบวนการคำนวณความเร็ว และกระบวนการเปลี่ยนตำแหน่งเท่านั้น ซึ่งถ้าเข้าใจขั้นตอนเหล่านี้ก็จะเข้าใจหลักการการทำงานของ PSO

PSO สามารถประยุกต์ใช้กับจำนวนจริง [10] เนื่องจาก PSO มีรูปแบบการเข้ารหัสที่เป็นจำนวนจริง และมีการคำนวณแบบเวกเตอร์ ทำให้ PSO สามารถประยุกต์ใช้กับจำนวนจริงได้อย่างมีประสิทธิภาพ

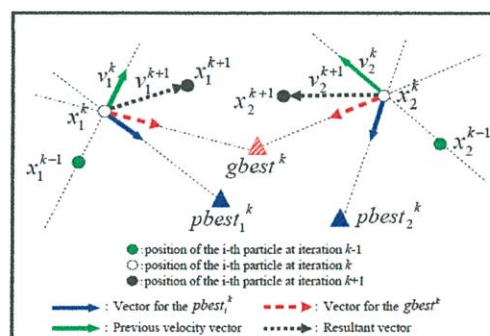
PSO สามารถประยุกต์ใช้แก้ปัญหาค้นหาจุดที่ดีที่สุดในงานวิจัยด้านวิทยาศาสตร์และวิศวกรรม [10] เนื่องจากงานด้านวิทยาศาสตร์และวิศวกรรม ส่วนมากมีรูปแบบเป็นจำนวนจริง ซึ่ง PSO จะมีประสิทธิภาพในการแก้ปัญหารูปแบบจำนวนจริง

สำหรับปัญหาค้นหาจุดที่ดีที่สุดหลายปัญหา เช่น ปัญหาฟังก์ชันการหาจุดที่ดีที่สุด (Function Optimization) [85-87] การฝึกอบรมเครือข่ายประสาทเทียม (Artificial Neural Network Training) [88, 89] การจัดกลุ่มและการควบคุมระบบซับซ้อน (Pattern Classification and Fuzzy System Control) [90, 91] ผลลัพธ์การค้นหาคำตอบของ PSO ได้รับคำตอบที่ดีกว่า และมีอัตราการลู่เข้าหาคำตอบที่เร็วกว่า [7-9] เมื่อเปรียบเทียบขั้นตอนวิธีที่มีอยู่ในกลุ่มของขั้นตอน

วิธีการหาค่าการหาจุดที่ดีที่สุดที่มีลักษณะพื้นฐานประชากรและมีการค้นหาอย่างไม่แน่นอน (Population-based Stochastic Optimization Algorithm) ตัวอย่างเช่น GA เป็นต้น

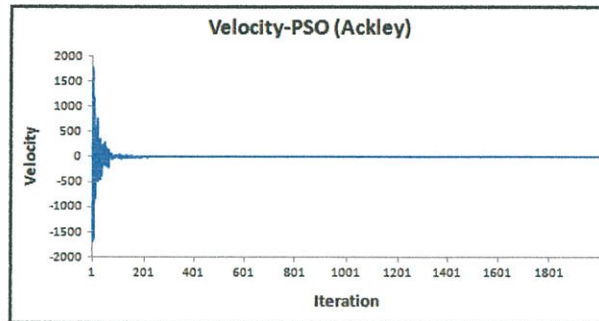
2.6.2 ข้อเสียของ PSO

สำหรับงานวิจัยบางงานได้อธิบายถึงสาเหตุที่ PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ง่าย เพราะผลจากขั้นตอนกระบวนการคำนวณความเร็วและกระบวนการเปลี่ยนตำแหน่ง [13] ผลกระทบจากสมการ 2.2 และพฤติกรรมการทำงานของ PSO [92] ผลจากอิทธิพลแรงดึงดูดของ GBEST [41] โดยสรุปคือ พฤติกรรมของ PSO อนุภาคทั้งหมดจะเคลื่อนที่เข้าหา GBEST ดังรูปที่ 2.13 เมื่อ GBEST ติดอยู่ในจุดสุดสัมพัทธ์แล้วไม่มีอนุภาคใดที่สามารถหาตำแหน่งที่ดีกว่า GBEST อนุภาคเหล่านี้ก็จะลู่เข้าหา GBEST อย่างรวดเร็ว เมื่ออนุภาคทั้งหมดเคลื่อนที่ไปหา GBEST ก็เสมือนเคลื่อนที่ไปติดจุดสุดสัมพัทธ์ นอกจากนี้อนุภาคที่เคลื่อนที่เข้าใกล้ GBEST เรื่อยๆ จะส่งผลให้ผลต่างระยะทางระหว่างอนุภาค PBEST และ GBEST มีค่าลดลงเรื่อยๆ จากสมการ 2.2 เมื่อผลต่างเหล่านี้ลดลงจะส่งผลให้ความเร็วของอนุภาคมีค่าลดลง และเมื่อความเร็วในรอบปัจจุบันมีค่าลดลง จะส่งผลให้รอบถัดไปมีความเร็วลดลง เพราะว่าความเร็วในรอบปัจจุบันจะถูกใช้ในการคำนวณในรอบถัดไป สามารถสังเกตได้จากกราฟ 2.14 ซึ่งมาจากการทดลอง 5.1 จะเห็นว่าความเร็วของอนุภาคในช่วงแรกจะมีค่ามากแล้วหลังจากนั้นจะมีค่าลดลงจนเกือบเป็นศูนย์ จากสมการ 2.3 เมื่อความเร็วลดลงส่งผลให้อนุภาคเคลื่อนที่ในบริเวณแคบลงเรื่อยๆ จึงทำให้ความหลากหลายของประชากรจะลดลงเรื่อยๆ ดังนั้นอนุภาคทั้งหมดจะค้นหาในบริเวณรอบจุดสุดสัมพัทธ์ตลอดไป และไม่ออกไปค้นหาในบริเวณอื่นอีก จึงไม่สามารถหาจุดสุดสัมบูรณ์พบ โดยเหตุการณ์ที่อนุภาคลู่เข้าไปติดอยู่ในจุดสุดสัมพัทธ์อย่างรวดเร็วนี้ถูกเรียกว่าการลู่เร็วเกินไป (Premature Convergence) โดยเหตุการณ์นี้ [71] จะเกิดขึ้นเมื่อความหลากหลายของประชากรลดลงอย่างรวดเร็ว สามารถสังเกตได้จากกราฟ 2.15 ซึ่งมาจากการทดลอง 5.1 จะเห็นว่าความหลากหลายของประชากรในช่วงแรกจะมีค่ามากแล้วหลังจากนั้นจะมีค่าลดลงแล้วก็คงที่ อย่างไรก็ตามระหว่างที่อนุภาคกำลังลู่เข้าหา GBEST มีบางอนุภาคสามารถพบค่าที่ดีกว่า GBEST ก็จะสามารถหลุดออกจากจุดสุดสัมพัทธ์ได้ แต่โอกาสที่อนุภาคจะพบตำแหน่งที่ดีกว่า GBEST นั้นเกิดขึ้นได้น้อยมาก ดังนั้น PSO จึงมีโอกาสติดในจุดสุดสัมพัทธ์ได้ง่ายซึ่งสามารถสังเกตได้จากตารางผลการทดลอง 5.4

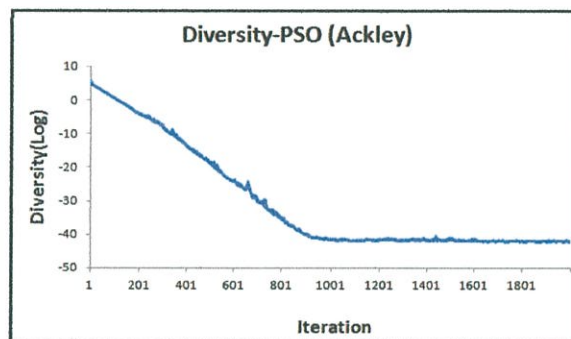


รูปที่ 2.13 พฤติกรรมของการเคลื่อนที่ของ PSO [92]

จากรูป 2.13 สมมติให้ปัญหาที่แก้ไชนั้นมี 2 มิติ โดยแต่ละอนุภาคนั้นจะมี 4 องค์ประกอบที่มีอิทธิพลต่อการเคลื่อนของอนุภาค คือตำแหน่งของอนุภาค ความเร็วของอนุภาค PBEST และ GBEST โดยในรูปนี้ได้มีการเขียนสัญลักษณ์แทนองค์ประกอบดังนี้ ตำแหน่งของอนุภาค คือ $X_1 = (X_{11}, X_{12}, \dots, X_{1m})$ ความเร็วของอนุภาค คือ $V_1 = (V_{11}, V_{12}, \dots, V_{1m})$ โดย m คือ จำนวนของอนุภาค และ n คือ จำนวนมิติของปัญหา จำนวนรอบการค้นหาคำตอบ คือ $k = 1, \dots, K$ โดย K คือ จำนวนรอบการค้นหาสูงสุด



รูปที่ 2.14 ความเร็วของ PSO กับปัญหาการติดในจุดสุดสัมพัทธ์



รูปที่ 2.15 ความหลากหลายของประชากรของ PSO กับปัญหาการติดในจุดสุดสัมพัทธ์

ความหลากหลายของประชากร ($D(t)$) สามารถคำนวณได้โดยใช้ความหลากหลายเฉลี่ยของระยะห่างระหว่างประชากร [93] โดย $D(t)$ สามารถใช้ในการอธิบายการกระจายตัวของกลุ่มอนุภาค ถ้า $D(t)$ มีค่าน้อย คือ ประชากรมีการกระจายตัวน้อยและค้นหาในบริเวณที่แคบ ในทางกลับกัน ถ้า $D(t)$ มีค่ามาก คือ ประชากรมีการกระจายตัวมากและค้นหาในบริเวณที่กว้าง โดย $D(t)$ มีรูปแบบดังสมการที่ 2.4

$$D(t) = \frac{1}{M \times L} \sum_{i=1}^M \sqrt{\sum_{d=1}^D (x_{id}^t - \bar{p}_{id})^2} \quad (2.4)$$

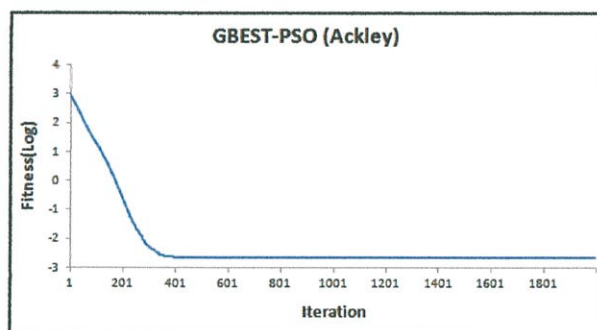
โดยกำหนดให้

- L คือ ความยาวของขอบเขตที่ต้องการค้นหาคำตอบ
- D คือ จำนวนมิติของการค้นหาคำตอบ
- M คือ ขนาดของประชากร
- x_{id}^t คือ ตำแหน่งของอนุภาคของอนุภาค i และมิติ d

\bar{p}_{id} คือ ค่าเฉลี่ยของอนุภาคทั้งหมดและมิติทั้งหมด

สำหรับ PSO เมื่อติดในจุดสุดสัมพัทธ์แล้ว GBEST จะไม่มีการเปลี่ยนแปลง ไม่ว่าจะดำเนินการค้นหาต่อไปอีกยาวนานแค่ไหนก็ตาม เนื่องจาก PSO ติดในจุดสุดสัมพัทธ์ได้ง่าย นอกจากนี้เมื่อเกิดปัญหานี้ขึ้นแล้ว PSO ไม่มีวิธีแก้ปัญหาค้นหาจุดสุดสัมพัทธ์ [77] จึงทำให้ PSO ไม่สามารถหาจุดสุดสัมบูรณ์ได้อย่างแน่นอน ดังนั้นเมื่อ PSO เกิดปัญหาค้นหาจุดสุดสัมพัทธ์ขึ้น ก็เสมือนว่าการค้นหาด้วย PSO นั้นสิ้นสุดลง การค้นหาต่อไปจะทำให้สิ้นเปลืองทรัพยากรโดยเปล่าประโยชน์ สามารถสังเกตได้จากกราฟ 2.16 ซึ่งมาจากการทดลอง 5.1 จะเห็นว่า GBEST มีค่าคงที่หลังจากการค้นหาไปสักระยะเวลาหนึ่ง แล้วไม่มีแนวโน้มที่จะดีขึ้นไม่ว่าจะค้นหาต่อไปอีกยาวนานแค่ไหนก็ตาม

อย่างไรก็ตามผลจากที่ GBEST ไม่มีเปลี่ยนแปลงเป็นเวลานานสามารถนำมาตรวจสอบการติดในจุดสุดสัมพัทธ์ของ PSO ได้ โดยสังเกตว่า GBEST ไม่มีการเปลี่ยนแปลงเป็นเวลานานๆ เช่น 100, 1000, 10000 รอบ เป็นต้น สามารถทำนายได้ว่าการติดในจุดสุดสัมพัทธ์ขึ้น ซึ่งระยะเวลาที่ใช้ในการตรวจสอบนั้น ขึ้นอยู่กับปัญหาที่นำไปประยุกต์ใช้ และเรียกจำนวนรอบนี้ว่าขอบเขตการเกิดจุดสุดสัมพัทธ์ (Local Optimum Threshold) ซึ่งจะกำหนดเอาไว้ล่วงหน้าเป็นค่าคงที่มักจะกำหนดเอาไว้มากๆ เพื่อที่จะตรวจสอบได้อย่างแม่นยำ



รูปที่ 2.16 GBEST ของ PSO กับปัญหาค้นหาจุดสุดสัมพัทธ์

แม้ว่า PSO มีแนวโน้มที่จะเกิดปัญหาค้นหาจุดสุดสัมพัทธ์ได้ง่าย แต่องค์ประกอบของการเกิดปัญหาค้นหาจุดสุดสัมพัทธ์ จากที่ได้กล่าวไปแล้วมี 2 องค์ประกอบ คือ ขั้นตอนวิธีที่นำมาแก้ปัญหาค้นหาจุดที่ดีที่สุดกับความซับซ้อนของปัญหาค้นหาจุดที่ดีที่สุด

สำหรับความซับซ้อนของปัญหาค้นหาจุดที่ดีที่สุดนั้น มีผลต่อการติดในจุดสุดสัมพัทธ์ของ PSO โดยถ้าปัญหายูนีโมดอล PSO จะไม่เกิดปัญหาค้นหาจุดสุดสัมพัทธ์และสามารถแก้ไขปัญหานี้ได้ดี เนื่องจากคุณสมบัติการลู่เข้าหาค่าตอบอย่างรวดเร็ว ทำให้พบคำตอบได้อย่างรวดเร็ว และสามารถหาค่าตอบที่ดีที่สุดได้อย่างแน่นอน ในทางกลับกันถ้าปัญหามัลติโมดอล PSO จะมีโอกาสมากที่จะติดในจุดสุดสัมพัทธ์ แต่อย่างไรก็ตาม PSO ก็เชื่อว่าแก้ปัญหาค้นหาจุดที่ดีที่สุดได้เช่นกัน ถ้าหากความซับซ้อนของปัญหาไม่ได้มากมายหรือจำนวนประชากรของ PSO เพียงพอที่จะแก้ปัญหาค้นหาจุดที่ดีที่สุดพบได้เช่นกัน

บทที่ 3

งานวิจัยที่เกี่ยวข้อง

จากบทที่ 2 จะเห็นว่า PSO มีประสิทธิภาพในการแก้ปัญหาการหาจุดที่ดีที่สุด แต่ PSO มีข้อเสียหลัก คือปัญหาการติดในจุดสุดสัมพัทธ์ เพื่อแก้ไขปัญหานี้จึงได้นำเทคนิคอื่นมาประยุกต์ใช้กับ PSO เช่น การกลายพันธุ์ (Mutation Operator) [11-47] การเริ่มต้นการค้นหาใหม่หรือรีรัน (RERUN) [48-52] การรวมกลุ่มของอนุภาคใหม่ (Regrouping) [95] และการปรับพารามิเตอร์ให้เหมาะสม [96] เป็นต้น

3.1 การประยุกต์การกลายพันธุ์กับ PSO

3.1.1 นิยามของการกลายพันธุ์

การกลายพันธุ์ (Mutation Operator) เป็นขั้นตอนวิธีที่นำมาจาก GA เพื่อให้เกิดการเปลี่ยนแปลงของประชากร ซึ่งเทียบเท่ากับการเคลื่อนที่ของอนุภาคของ PSO โดยอาศัยหลักการจากทฤษฎีวิวัฒนาการจากชีววิทยาและการคัดเลือกตามธรรมชาติ โดยสิ่งมีชีวิตจะมีพฤติกรรมการหาอาหาร เช่น การสืบพันธุ์ การกลายพันธุ์ การแลกเปลี่ยนยีน ทำให้เกิดสิ่งมีชีวิตพันธุ์ใหม่ โดยสิ่งมีชีวิตที่เหมาะสมที่สุดจึงจะอยู่รอดมาได้ กระบวนการคัดเลือกนี้ได้เปลี่ยนแปลงสิ่งมีชีวิตให้เหมาะสมยิ่งขึ้น

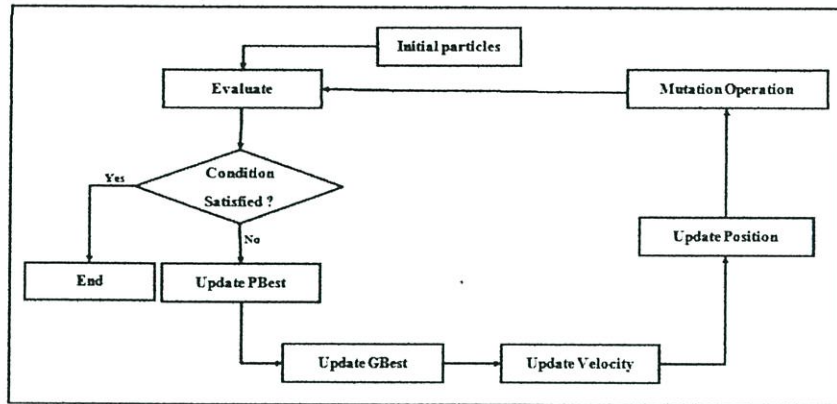
โดยหลักการของการกลายพันธุ์ คือการสุ่มเปลี่ยนตำแหน่งของอนุภาคใดๆ ภายในกลุ่มของประชากร โดยเปลี่ยนค่าในบางมิติของอนุภาคใดๆ ในจำนวนเพียงเล็กน้อย เช่น 1%, 2%, 3%, 5% จากจำนวนมิติทั้งหมดภายในประชากร เป็นต้น โดยจะเรียกค่านี้ว่า ความน่าจะเป็นในการกลายพันธุ์ (P_m หรือ Mutation Probability) ผลจากการกลายพันธุ์ทำให้ตำแหน่งเปลี่ยนแปลงจากเดิมไม่มากนัก

งานวิจัย [97] กล่าวว่าสภาพที่ PSO มีความหลากหลายของประชากรน้อยนั้น คือสภาพที่อนุภาคติดอยู่ในจุดสุดสัมพัทธ์ ดังนั้นเพื่อแก้ปัญหาการติดอยู่ในจุดสุดสัมพัทธ์ จึงต้องเพิ่มความหลากหลายของประชากร สำหรับงานวิจัย [98] กล่าวว่า การกลายพันธุ์สามารถเพิ่มการกระจายตัวและความหลากหลายของประชากร จึงทำให้การค้นหาที่มีบริเวณที่กว้างขึ้นและป้องกันอนุภาคเข้าสู่หาจุดสุดสัมพัทธ์ สำหรับงานวิจัย [12, 92] กล่าวว่า การกลายพันธุ์เพิ่มโอกาสให้อนุภาคกระโดดออกจากจุดสุดสัมพัทธ์ จึงเพิ่มประสิทธิภาพการทำงานของ PSO และงานวิจัย [12] กล่าวว่า การกลายพันธุ์ทำให้เกิดการเปลี่ยนแปลงของประชากร และทำลายโครงสร้างของประชากรที่สืบทอดมาจากรุ่นก่อนหน้าส่งผลเกิดการค้นหามีบริเวณใหม่ นอกจากนี้การกลายพันธุ์สามารถนำมาประยุกต์ใช้กับ PSO ได้ง่าย จึงเป็นวิธีที่นิยมนำมาประยุกต์ใช้เพื่อแก้ปัญหาการติดในจุดสุดสัมพัทธ์

จากผลการทดลองของตาราง 5.5 และ 5.6 ซึ่งมาจากผลการทดลองในหัวข้อ 5.2 แสดงให้เห็นว่าการกลายพันธุ์สามารถเพิ่มประสิทธิภาพการค้นหาของ PSO โดยภาพรวมแล้ว PSO ที่มีขั้นตอนการประยุกต์การกลายพันธุ์จะได้รับคำตอบดีกว่า PSO ที่ไม่มีขั้นตอนการประยุกต์การกลายพันธุ์

ด้วยเหตุที่กล่าวมานี้ นักวิจัยจำนวนมาก [11-47] จึงได้นำการกลายพันธุ์เพิ่มเข้าไปในขั้นตอนการทำงานของ PSO

3.1.2 ขั้นตอนของการกลายพันธุ์



รูปที่ 3.1 แผนภาพขั้นตอนการประยุกต์การกลายพันธุ์กับ PSO

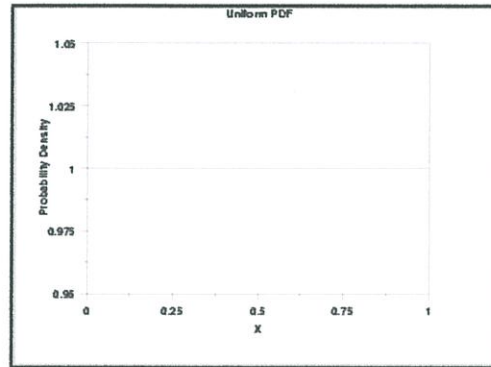
ขั้นตอนการทำงานของการกลายพันธุ์ถูกแสดงดังรูป 3.1 โดยจะเริ่มจากการสร้างอนุภาคขึ้นมาโดยสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัวภายในปริภูมิของคำตอบสำหรับจำนวนอนุภาคที่สร้างขึ้นนี้จะไม่มีการเพิ่มหรือลดจำนวนตลอดการทำงาน จากนั้นเป็นขั้นตอนหาค่าความเหมาะสม โดยนำอนุภาคทั้งหมดไปหาค่าความเหมาะสมจากตำแหน่งปัจจุบันของอนุภาคด้วยฟังก์ชันค่าความเหมาะสม ในขั้นตอนต่อมาถ้าอนุภาคสามารถพบคำตอบของปัญหาหรือครบจำนวนรอบที่กำหนดเอาไว้ก็จะจบการค้นหา แต่ถ้าไม่พบหรือไม่ครบก็จะดำเนินการในขั้นตอนถัดไปคือ ปรับปรุง PBest ปรับปรุง GBest ปรับปรุงความเร็ว และปรับปรุงตำแหน่งของอนุภาคตามลำดับ โดยขั้นตอนเหล่านี้จะทำงานเช่นเดียว PSO สำหรับขั้นตอนต่อมา คือขั้นตอนการกลายพันธุ์ โดยขั้นตอนนี้จะทำการตรวจสอบเงื่อนไขของการกลายพันธุ์ ถ้าตรงตามเงื่อนไขที่กำหนด เช่นเงื่อนไขกำหนดให้ดำเนินการกลายพันธุ์ทุกสิกรอบของรอบการค้นหาคำตอบ เป็นต้น ถ้าตามเงื่อนไขจะดำเนินการกลายพันธุ์ แล้ววนกลับไปขั้นตอนหาค่าความเหมาะสมของอนุภาค

3.1.3 องค์ประกอบของการกลายพันธุ์

3.1.3.1 ค่าสุ่มจากการกระจายตัว

ค่าสุ่มจากการกระจายตัว (Distribution Probability) คือค่าสุ่มที่ได้มาจากการกระจายตัวแบบต่างๆ เพื่อเปลี่ยนค่าของมิติบนตำแหน่งหรือความเร็ว โดยค่าสุ่มนี้จะถูกนำมาประยุกต์ใช้กับสมการกลายพันธุ์ ซึ่งค่าสุ่มนี้มีหลายแบบและแต่ละแบบส่งผลให้เกิดการกระจายตัวและความหลากหลายของประชากรที่แตกต่างกัน ตัวอย่างค่าสุ่มจากการกระจายตัวที่ถูกนำมาใช้ในงานวิจัยต่างๆ มีดังนี้

- งานวิจัย [84] ใช้การกระจายเอกรูป (Uniform Distribution) คือการแจกแจงแบบสม่ำเสมอในช่วงการกระจายตัว $[a, b]$ จะได้มัธยฐาน (Median) เท่ากับ $(a + b) / 2$ ซึ่งมีค่าเท่ากับมัชฌิมา (Mean) ดังแสดงในรูป 3.2 และสมการ 3.1



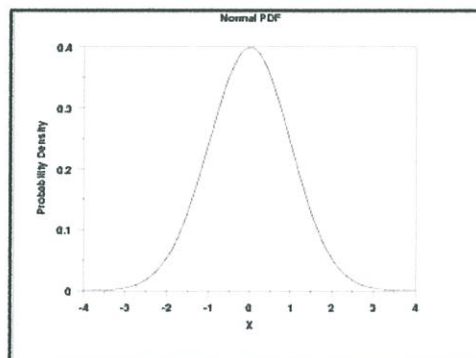
รูปที่ 3.2 การกระจายเอกรูป

$$f(x) = Rand(x) \quad (3.1)$$

โดยกำหนดให้

$Rand(x)$ คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกกำหนด

- งานวิจัย [14] และ [18] ใช้การกระจายตัวปกติ (Normal distribution หรือ Gaussian distribution) คือการกระจายตัวที่มีมัชฌิมา (μ) เท่ากับมัธยฐานเท่ากับฐานนิยม (Mode) ดังแสดงในรูป 3.3 และสมการ 3.2



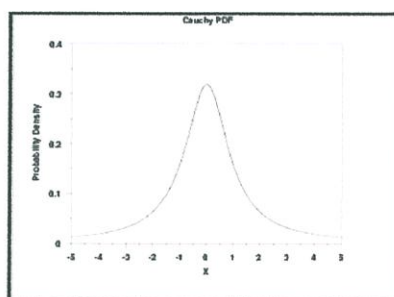
รูปที่ 3.3 การกระจายตัวปกติ

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad (3.2)$$

โดยกำหนดให้

- a คือ ค่าคงที่ที่เป็นจำนวนจริงซึ่งมีค่ามากกว่า 0
- b คือ ค่าคงที่ที่เป็นจำนวนจริงซึ่งมีค่ามากกว่า 0
- c คือ ค่าคงที่ที่เป็นจำนวนจริงซึ่งมีค่ามากกว่า 0
- e คือ ค่า Euler's number เป็นค่าคงที่มีค่าประมาณ 2.71828
- x คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกกำหนด

- งานวิจัย [19] ใช้การกระจายตัวโคซี (Cauchy Distribution) คือการกระจายตัวที่มีพารามิเตอร์บ่งชี้ตำแหน่ง x_0 และพารามิเตอร์บ่งชี้ขนาด y โดยมีมาตรฐานจะมีค่าเท่ากับ x_0 ดังแสดงในรูป 3.4 และสมการ 3.3 การแจกแจงโคซีมีรูปแบบเหมือนกับการแจกแจงปกติ แต่ต่างตรงที่ขอบทั้งสองด้านมีโอกาสกระจายตัวมากกว่าการแจกแจงปกติ ดังนั้นค่าที่ได้จากการกระจายตัวแบบโคซีจะมีค่ากว้างกว่าค่าที่ได้กระจายตัวแบบปกติ



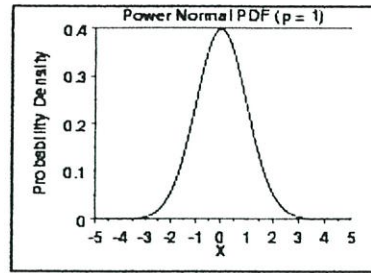
รูปที่ 3.4 การกระจายตัวโคซี

$$f(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2} \quad (3.3)$$

โดยกำหนดให้

- t คือ ค่ามาตรฐานการกระจายตัวแบบโคซี
- x คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกกำหนด

- งานวิจัย [17] และ [38] ใช้การกระจายยกกำลัง (Power Distribution) คือการกระจายตัวแบบยกกำลังในช่วง $[a, b]$ ดังแสดงในรูป 3.5 และสมการ 3.4 โดยค่าที่ได้จะมีค่ามากหรือน้อยขึ้นอยู่กับเลขชี้กำลัง ถ้าเลขชี้กำลังมีค่ามากผลลัพธ์ที่ได้ก็จะมีค่ามาก



รูปที่ 3.5 การกระจายยกกำลัง

$$f(x) = (x)^p \quad (3.4)$$

โดยกำหนดให้

- p คือ เลขชี้กำลัง
- x คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกกำหนด

3.1.3.2 สมการการกลายพันธุ์

สมการการกลายพันธุ์ (Mutation Equation) คือสมการที่ประยุกต์ใช้ในขั้นตอนการกลายพันธุ์ โดยสมการนี้นำมาประยุกต์ใช้กับมิติบนตำแหน่งหรือความเร็ว เพื่อให้เกิดการเปลี่ยนค่าของตำแหน่งหรือความเร็ว ซึ่งทำให้เกิดความหลากหลายและการกระจายตัวของประชากร สำหรับการออกแบบสมการการกลายพันธุ์ อาจจะมีรูปแบบและค่าสุ่มจากการกระจายตัวเหมือนกันหรือแตกต่างกันขึ้นอยู่กับผู้ออกแบบสมการการกลายพันธุ์ ซึ่งแต่ละวิธีมีผลลัพธ์ที่ต่างกัน โดยเอกสาร [99] ได้กล่าวว่าการกลายพันธุ์แบบกระจายตัวโคซีเหมาะกับการแก้ไขปัญหามัลติโมดอล ในขณะที่การกลายพันธุ์แบบกระจายยกกำลังเหมาะกับการแก้ปัญหาไนโมดอล สำหรับตัวอย่างสมการการกลายพันธุ์ที่ถูกนำมาใช้ในงานวิจัยต่างๆ มีดังนี้

- งานวิจัย [14] และ [18] ใช้สมการการกลายพันธุ์ที่ใช้กระจายตัวปกติ จึงเรียกรวีนีว่าการกลายพันธุ์แบบปกติ (Gaussian Mutation) โดยภายในหนึ่งมิติของอนุภาค จะดำเนินการการกลายพันธุ์โดยใช้สมการ 3.5 โดยกำหนดให้ σ มีค่าเท่ากับ 0.1 ซึ่งเป็นพารามิเตอร์ที่ใช้กำหนดรูปแบบการกระจายตัวของกราฟ

$$mutate(x_{id}) = x_{id}(1 + Gaussian(\sigma)) \quad (3.5)$$

โดยกำหนดให้

- Gaussian (σ) คือ ค่าที่สุ่มจากการกระจายตัวปกติ
- σ คือ ส่วนเบี่ยงเบนมาตรฐานซึ่งเป็นค่าคงที่

- งานวิจัย [19] ใช้สมการการกลายพันธุ์ที่ใช้การกระจายตัวแบบโคชี จึงเรียกรวีนี้อีกว่า การกลายพันธุ์แบบโคชี (Cauchy Mutation) โดยภายในหนึ่งมิติของอนุภาค จะดำเนินการการกลายพันธุ์โดยใช้สมการ 3.6 โดยกำหนดให้ δ มีค่าเท่ากับ 0.2 ซึ่งเป็นพารามิเตอร์ที่ใช้กำหนดรูปแบบการกระจายตัวของกราฟ

$$\text{mutate}(x_{id}) = x_{id} + \text{Cauchy}(\delta) \quad (3.6)$$

โดยกำหนดให้

Cauchy(δ)	คือ ค่าที่สุ่มจากการกระจายตัวแบบโคชี
δ	คือ ส่วนเบี่ยงเบนมาตรฐานซึ่งเป็นค่าคงที่

- งานวิจัย [17] และ [38] ใช้กระจายยกกำลัง จึงเรียกรวีนี้อีกว่าการกลายพันธุ์แบบยกกำลัง (Power Mutation) โดยภายในหนึ่งมิติของอนุภาค จะทำการกลายพันธุ์โดยใช้สมการ 3.7

$$f(x) = \begin{cases} \bar{x} - s(\bar{x} - x') & t < r \\ \bar{x} + s(x'' - \bar{x}) & t \geq r \end{cases} \quad (3.7)$$

โดยกำหนดให้

t	คือ $\frac{\bar{x} - x'}{x'' - \bar{x}}$
x'	คือ ขอบเขตล่างที่ใช้กำหนดค่าของตัวแปร
x''	คือ ขอบเขตบนที่ใช้กำหนดค่าของตัวแปร
r	คือ ตัวเลขจำนวนจริงที่ทำการสุ่มอยู่ในช่วง 0 ถึง 1
\bar{x}	คือ ค่าเฉลี่ยของตัวแปร
$s(x)$	คือ x^p
p	คือ เลขชี้กำลัง
x	คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกกำหนด

- งานวิจัย [20] ใช้การกลายพันธุ์แบบโคชี โดยการกระจายตัวแบบโคชีได้นำมาประยุกต์ใช้กับความเร็วและตำแหน่งของอนุภาค ซึ่งแตกต่างจากการกลายพันธุ์แบบโคชีของงานวิจัย [19] โดยการกลายพันธุ์วิธีนี้ใช้สมการ 3.8

$$\begin{aligned} V'_{id} &= V_{id} e^{\text{Cauchy}(\delta)} \\ X'_{id} &= X_{id} + V'_{id} \text{Cauchy}(\delta) \end{aligned} \quad (3.8)$$

โดยกำหนดให้

Cauchy(δ) คือ ค่าที่สุ่มจากกระจายตัวแบบโคชี
 δ คือ ส่วนเบี่ยงเบนมาตรฐานซึ่งเป็นค่าคงที่

- งานวิจัย [53] ใช้สมการการกลายพันธุ์ที่ใช้การกระจายตัวแบบมิเชลวิซ (Michalewicz Distribution) ซึ่งวิธีนี้แบ่งสมการการกลายพันธุ์เป็น 2 แบบ คือ มิเชลวิซแบบไม่มีเอกรูป (Non-uniform Michalewicz Operator) โดยมีรูปแบบดังสมการ 3.9 และ 3.10 มิเชลวิซแบบมีเอกรูป (Uniform Michalewicz Operator) โดยมีรูปแบบดังสมการ 3.9 และ 3.11

$$\text{mutate}(x_{id}) = x_{id} + \text{delta}(t, U - x_{id}) : rb = 1 \quad (3.9)$$

$$\text{mutate}(x_{id}) = x_{id} + \text{delta}(t, x_{id} - L) : rb = 0$$

$$\text{delta}(t, y) = y(1 - r^{(1-t/T)^b}) \quad (3.10)$$

$$\text{delta}(t, y) = y(1 - r) \quad (3.11)$$

โดยกำหนดให้

t คือ สถานะเวลาของรอบปัจจุบัน
 U คือ ขอบเขตบนสุดของค่าของอนุภาคของแต่ละมิติ
 L คือ ขอบเขตล่างสุดของอนุภาคของแต่ละมิติ
 rb คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกระบุ
 delta(t, y) คือ ค่าที่ได้จากค่าที่อยู่ในช่วงของ [0 : y]
 r คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วง [0, 1]
 T คือ สถานะเวลาของรอบทั้งหมดในการค้นหาคำตอบ
 b คือ ค่าคงที่ค่าหนึ่งที่เป็นจำนวนจริง

- งานวิจัย [40] ใช้สมการการกลายพันธุ์ที่ใช้กระจายตัวแบบโคชี กระจายตัวปกติ และการกระจายตัวแบบเลวี (Levy Distribution) ตามลำดับ สมการเหล่านี้ถูกนำมาประยุกต์ใช้กับความเร็ว และ GBEST ซึ่งแตกต่างจากสมการของ [19] และ [20] โดยมีรูปแบบดังสมการ 3.12, 3.13, 3.14 ตามลำดับ

$$\vec{v}'_g = \vec{v}_g e^{\text{Cauchy}(\delta)} \quad (3.12)$$

$$\vec{x}'_g = \vec{x}_g + \vec{v}'_g \text{Cauchy}(\delta)_g$$

$$\vec{v}'_g = \vec{v}_g e^{\text{Gaussian}(\sigma)} \quad (3.13)$$

$$\vec{x}'_g = \vec{x}_g + \vec{v}'_g \text{Gaussian}(\sigma)_g$$

$$\vec{v}'_g = \vec{v}_g e^{\text{Levy}()_g} \quad (3.14)$$

$$\vec{x}'_g = \vec{x}_g + \vec{v}'_g \text{Levy}()_g$$

โดยกำหนดให้

\bar{x}_g	คือ ตำแหน่งของอนุภาคที่ดีที่สุด
\bar{v}_g	คือ ความเร็วของอนุภาคที่ดีที่สุด
Cauchy (δ)	คือ ค่าที่สุ่มจากกระจายตัวแบบโคชี
Cauchy(δ) _g	คือ ค่าที่สุ่มจากกระจายตัวแบบโคชีของอนุภาคที่ดีที่สุด
δ	คือ ส่วนเบี่ยงเบนมาตรฐานซึ่งเป็นค่าคงที่
Gaussian (σ)	คือ ค่าที่สุ่มจากกระจายตัวปกติ
Gaussian(σ) _g	คือ ค่าที่สุ่มจากกระจายตัวปกติ
σ	คือ ส่วนเบี่ยงเบนมาตรฐานซึ่งเป็นค่าคงที่
Levy()	คือ ค่าที่สุ่มจากกระจายตัวแบบเลวี
Levy() _g	คือ ค่าที่สุ่มจากกระจายตัวแบบเลวีของอนุภาคที่ดีที่สุด

สำหรับวิทยานิพนธ์ฉบับนี้ ใช้สมการการกลายพันธุ์ที่ใช้การกระจายตัวแบบเอกรูป โดยรูปแบบสมการการกลายพันธุ์มีดังนี้ ถ้าค่าที่สุ่มตัวเลขจำนวนจริงในขอบเขตระหว่าง 0 กับ 1 มีค่าน้อยกว่า 0.5 ให้ดำเนินการกลายพันธุ์ในตำแหน่งของอนุภาคและมิติอื่นๆ โดยใช้สมการ 3.15 นอกจากนี้ให้ดำเนินการกลายพันธุ์ในตำแหน่งของอนุภาคและมิติอื่นๆ โดยใช้สมการ 3.16

$$\text{mutate}(x_{id}) = x_{id} - x_{id} \times \text{Rand}() \quad (3.15)$$

$$\text{mutate}(x_{id}) = x_{id} + x_{id} \times \text{Rand}() \quad (3.16)$$

โดยกำหนดให้

x_{id}	คือ ตำแหน่งของอนุภาคที่ตำแหน่งของ i อนุภาค และ d มิติ
Rand()	คือ ตัวเลขที่ทำการสุ่มอยู่ในช่วงขอบเขตที่ถูกกำหนด

3.1.3.3 พารามิเตอร์ที่ถูกดำเนินการการกลายพันธุ์

พารามิเตอร์ที่ถูกดำเนินการการกลายพันธุ์ (Mutated Parameter) คือ ตำแหน่งหรือความเร็วของ PSO ที่ถูกดำเนินการกลายพันธุ์ สำหรับตำแหน่งที่ถูกดำเนินการกลายพันธุ์ (Mutated Position) ประกอบด้วย 3 ส่วน คือ PBEST GBEST ตำแหน่งของอนุภาค และสำหรับความเร็วที่ถูกดำเนินการกลายพันธุ์ (Mutated Velocity) คือความเร็วของอนุภาค สำหรับการกำหนดพารามิเตอร์ที่ถูกดำเนินการการกลายพันธุ์ ขึ้นอยู่กับผู้ออกแบบขั้นตอนวิธีซึ่งจะให้ผลลัพธ์ที่แตกต่าง สำหรับตัวอย่างพารามิเตอร์ที่ถูกดำเนินการการกลายพันธุ์ที่ถูกนำมาใช้ในงานวิจัยต่างๆ มีดังนี้ งานวิจัย [20] ประยุกต์การกลายพันธุ์กับความเร็วและตำแหน่งของอนุภาค งานวิจัย [21] และ [41] ประยุกต์การกลายพันธุ์กับ GBEST งานวิจัย [46] และ [45] ประยุกต์การกลายพันธุ์กับ PBEST งานวิจัย [11, 13, 21, 33] นำการกลายพันธุ์มาประยุกต์ใช้กับตำแหน่งของอนุภาค

3.1.3.4 รอบที่จะดำเนินการกลายพันธุ์

รอบที่จะดำเนินการกลายพันธุ์ (Mutation Period) คือเงื่อนไขที่ใช้กำหนดว่าเมื่อใดดำเนินการกลายพันธุ์ โดยรอบที่จะดำเนินการกลายพันธุ์นั้น กำหนดได้หลายวิธี สำหรับตัวอย่างงานวิจัยมีดังนี้

- รอบที่จะดำเนินการกลายพันธุ์ของวิธีอัตราการกลายพันธุ์คงที่ [16] โดยรอบที่จะดำเนินการกลายพันธุ์จะถูกดำเนินการในทุกรอบของการค้นหาคำตอบ
- รอบที่จะดำเนินการกลายพันธุ์ของวิธีลดลงตามเชิงเส้น [16] โดยรอบที่จะดำเนินการกลายพันธุ์จะถูกดำเนินการในทุกรอบของการค้นหาคำตอบ
- รอบที่จะดำเนินการกลายพันธุ์ของวิธีเมื่อคำตอบคงที่ [16] การกลายพันธุ์จะถูกดำเนินการเมื่อคำตอบคงที่เป็นระยะเวลาเท่ากับค่าที่กำหนดของการกลายพันธุ์ (Mutation Threshold หรือ Mutation Period) หลังจากดำเนินการกลายพันธุ์แล้ว ถ้าคำตอบไม่มีการเปลี่ยนก็จะทำการกลายพันธุ์ไปเรื่อยๆ แต่ถ้าคำตอบมีการเปลี่ยนก็จะหยุดการกลายพันธุ์ และเริ่มนับจำนวนรอบที่ซ้ำใหม่
- รอบที่จะดำเนินการกลายพันธุ์ของวิธีรอบการค้นหาคำตอบที่คงที่ [17] การกลายพันธุ์พิจารณาจากรอบการค้นหาคำตอบ โดยจะกำหนดค่าคงที่รอบการค้นหา (Iteration Threshold หรือ Mutation Period) ซึ่งเป็นค่าคงที่ เมื่อรอบการค้นหาคำตอบหารด้วยค่าคงที่รอบการค้นหาลงตัวก็จะดำเนินการกลายพันธุ์
- รอบที่จะดำเนินการกลายพันธุ์ของวิธีความหนาแน่นของอนุภาค [13-15] การกลายพันธุ์จะพิจารณาจากความหนาแน่นของประชากร ถ้าค่าเฉลี่ยระยะห่างของประชากร (Population average distance) ($D(t)$) [93] มีค่าน้อยกว่า ค่าสถานะความหนาแน่นของประชากรน้อยสุด (D_{min}) หรือ GBEST ไม่มีการเปลี่ยนแปลงเป็นระยะเวลายาวนานเท่ากับค่าที่กำหนดให้ (Iteration Threshold) ก็จะดำเนินการกลายพันธุ์

3.1.3.5 จำนวนการกลายพันธุ์

จำนวนการกลายพันธุ์ (Mutation Amount) คือจำนวนมิติบนอนุภาคที่ถูกดำเนินการกลายพันธุ์ในแต่ละรอบที่จะดำเนินการกลายพันธุ์ โดยจำนวนมิติบนอนุภาคที่จะถูกดำเนินการกลายพันธุ์นี้จะมากหรือน้อยนั้นขึ้นอยู่กับค่าความน่าจะเป็นในการกลายพันธุ์ (Mutation Probability) ซึ่งค่าความน่าจะเป็นในการกลายพันธุ์จะเป็นตัวกำหนดโอกาสสุ่มมิติใดบนอนุภาคใดที่จะถูกดำเนินการกลายพันธุ์ โดยถ้าค่าที่สุ่มระหว่าง 0 กับ 1 มีค่าน้อยกว่าค่านี้ ก็จะดำเนินการกลายพันธุ์ สำหรับการกำหนดค่านี้ขึ้นอยู่กับผู้ออกแบบขั้นตอนวิธีและมักจะมีค่าเพียงเล็กน้อย เช่น

0.0001%, 0.001%, 0.1%, 0.5%, 1%, 5% เป็นต้น สำหรับจำนวนการกลายพันธุ์จะถูกกำหนดต่างกันตามวิธีที่นักวิจัยได้กำหนดไว้ สำหรับตัวอย่างงานวิจัยมีดังนี้

- งานวิจัย [16] ได้เสนอ วิธีการกลายพันธุ์แบบอัตราการกลายพันธุ์คงที่ โดยจำนวนการกลายพันธุ์ถูกพิจารณาจากอัตราการกลายพันธุ์ (Mutation Rate) ที่ถูกกำหนดไว้คงที่ โดยแต่ละมิติของอนุภาคจะถูกพิจารณาให้ดำเนินการกลายพันธุ์ ถ้าค่าที่สุ่มระหว่าง 0 กับ 1 มีค่าน้อยกว่า ความน่าจะเป็นในการกลายพันธุ์ ก็จะดำเนินการกลายพันธุ์ ณ มิติของอนุภาคนั้น สำหรับความน่าจะเป็นในการกลายพันธุ์นั้นขึ้นอยู่กับจำนวนของอนุภาค จำนวนของมิติของแต่ละอนุภาค และอัตราการกลายพันธุ์ซึ่งเป็นค่าคงที่ค่าหนึ่งที่ถูกกำหนดให้เป็นขอบเขตของการกลายพันธุ์ในแต่ละรอบ โดยสมการความน่าจะเป็นในการกลายพันธุ์ มีรูปแบบดังสมการ 3.17

$$\text{Mutation probability } y = \frac{\text{Mutation rate}}{\text{number of Particles} \times \text{number of Dimensions}} \quad (3.17)$$

- งานวิจัย [16] ได้เสนอจำนวนการกลายพันธุ์ลดลงตามเชิงเส้น คืออัตราการกลายพันธุ์จะแปรผกผันกับรอบของการค้นหาคำตอบแบบเชิงเส้น โดยการกลายพันธุ์จะใช้วิธีเดียวกับวิธีการกลายพันธุ์แบบอัตราการกลายพันธุ์คงที่ แต่ในช่วงแรกของการค้นหาคำตอบจะมีอัตราการกลายพันธุ์สูง เมื่อเวลาผ่านไปอัตราการกลายพันธุ์จะลดลงแบบเชิงเส้น โดยอัตราการกลายพันธุ์ ณ เวลาใดเวลาหนึ่งจะมีค่าเท่ากับผลลัพธ์จากสมการ 3.18

$$f(t) = \frac{T-t}{T} \times \text{Mutation_rate} \quad (3.18)$$

โดยกำหนดให้

- | | |
|---|-------------------------------------|
| T | คือ จำนวนรอบการค้นหาคำตอบสูงสุด |
| t | คือ จำนวนรอบปัจจุบันในการค้นหาคำตอบ |

- งานวิจัย [16] ได้เสนอจำนวนการกลายพันธุ์ของวิธีการกลายพันธุ์เมื่อคำตอบคงที่ คือ การกลายพันธุ์จะกระทำในช่วงที่คำตอบซ้ำกัน โดยการกลายพันธุ์จะใช้วิธีเดียวกับวิธีการกลายพันธุ์แบบอัตราการกลายพันธุ์คงที่
- งานวิจัย [17] ได้เสนอจำนวนการกลายพันธุ์จะถูกพิจารณาจากค่าคงที่ความน่าจะเป็นในการกลายพันธุ์ซึ่งเป็นค่าที่ถูกกำหนดให้ โดยในแต่ละมิติของอนุภาคแต่ละตัว ถ้าค่าที่สุ่มระหว่าง 0 กับ 1 มีค่าน้อยกว่าความน่าจะเป็นในการกลายพันธุ์ ก็จะดำเนินการกลายพันธุ์ ณ มิติของอนุภาคนั้น

- งานวิจัย [13] และ [14] ได้เสนอ จำนวนการกลายพันธุ์จะเกิดจากความน่าจะเป็นในการกลายพันธุ์ที่มีค่าสุ่มระหว่าง 0.1 ถึง 0.3 โดยในแต่ละมิติของอนุภาคแต่ละตัว ถ้าค่าที่สุ่มระหว่าง 0 กับ 1 มีค่าน้อยกว่าความน่าจะเป็นในการกลายพันธุ์ ก็จะมีการกลายพันธุ์ ณ มิติของอนุภาคนั้น

3.1.4 ประเภทของการกลายพันธุ์

ประเภทของการกลายพันธุ์สามารถแบ่งได้หลายประเภท ขึ้นอยู่ว่าจะเลือกพิจารณาองค์ประกอบส่วนไหนมาเป็นเกณฑ์ในการแบ่ง แต่สำหรับวิทยานิพนธ์ฉบับนี้เลือกพิจารณาจากการยอมให้ผลจากการกลายพันธุ์ทำให้ค่าความเหมาะสมแย่งลงหรือไม่ ซึ่งสามารถแบ่งออกเป็น 2 ประเภท [84] ดังนี้

3.1.4.1 การกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ

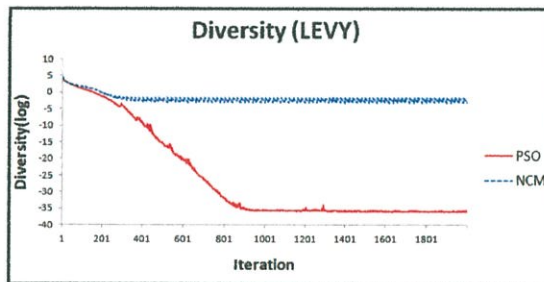
การกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ (Non-compared mutation หรือ NCM) [11-33] คือตำแหน่งจากการกลายพันธุ์จะนำแทนตำแหน่งที่ถูกดำเนินการการกลายพันธุ์ โดยไม่สนใจว่าตำแหน่งจากการกลายพันธุ์นั้น จะอยู่ในตำแหน่งที่ดีหรือแยกว่าตำแหน่งที่ถูกดำเนินการการกลายพันธุ์ โดยมีตัวอย่างงานวิจัยที่เลือกใช้วิธีนี้ เช่น งานวิจัย [11, 14, 16, 18] ได้ดำเนินการกลายพันธุ์กับตำแหน่งของอนุภาค ตำแหน่งที่ได้จากการกลายพันธุ์จะนำแทนที่ตำแหน่งของอนุภาคที่ถูกดำเนินการกลายพันธุ์ เป็นต้น

3.1.4.2 การกลายพันธุ์ประเภทมีการเปรียบเทียบ

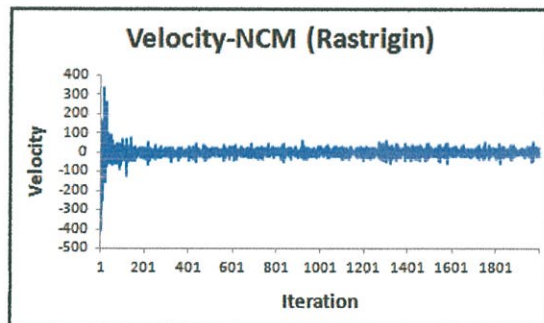
การกลายพันธุ์ประเภทมีการเปรียบเทียบ (Compared Mutation หรือ CM) [34 - 47] คือ ถ้าตำแหน่งจากการกลายพันธุ์อยู่ในตำแหน่งที่ดีกว่าตำแหน่งที่ถูกดำเนินการกลายพันธุ์ ตำแหน่งที่ได้จากการกลายพันธุ์จะนำไปแทนตำแหน่งที่ถูกดำเนินการกลายพันธุ์ แต่ถ้าตำแหน่งจากการกลายพันธุ์อยู่ในตำแหน่งที่แยกว่าตำแหน่งที่ถูกดำเนินการกลายพันธุ์ ตำแหน่งที่ได้จากการกลายพันธุ์จะไม่ถูกนำมาใช้ สำหรับวิธีนี้ในบางเอกสารได้เรียกวิธีนี้ว่าการค้นหาคำตอบเฉพาะที่ (Local Search) [41-43] โดยมีตัวอย่างงานวิจัยที่เลือกใช้วิธีนี้ เช่น งานวิจัย [45, 46] ได้ดำเนินการกลายพันธุ์กับ PBEST ถ้าตำแหน่งที่ได้จากการกลายพันธุ์อยู่ให้ตำแหน่งที่ดีกว่า PBEST ตำแหน่งที่ได้จากการกลายพันธุ์จะนำไปแทน PBEST แต่ถ้าตำแหน่งที่ได้จากการกลายพันธุ์อยู่ในตำแหน่งที่แยกว่า PBEST ก็จะไม่นำตำแหน่งที่ได้จากการกลายพันธุ์ไปแทน PBEST, งานวิจัย [34-42] ได้ดำเนินการกลายพันธุ์กับ GBEST ถ้าตำแหน่งที่ได้จากการกลายพันธุ์อยู่ที่ตำแหน่งซึ่งดีกว่า GBEST ตำแหน่งที่ได้จากการกลายพันธุ์จะนำไปแทน GBEST แต่ถ้าตำแหน่งที่ได้จากการกลายพันธุ์อยู่ในตำแหน่งแยกว่า GBEST ก็จะไม่นำตำแหน่งที่ได้จากการกลายพันธุ์ไปแทน GBEST เป็นต้น

3.1.5 ข้อดีของการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ

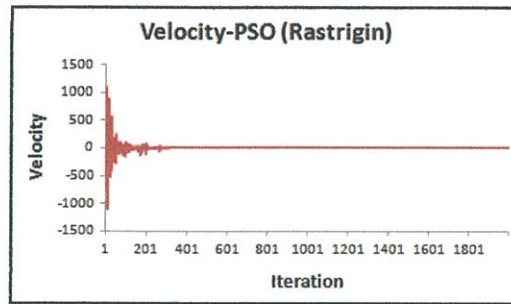
จากที่ได้กล่าวก่อนหน้านี การประยุกต์การกลายพันธุ์กับ PSO สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์และได้รับผลลัพธ์การค้นหาที่ดีขึ้น โดยการกลายพันธุ์ได้เพิ่มความหลากหลายของประชากร เช่นเดียวกันการกลายพันธุ์ประเภทนี้สามารถเพิ่มความหลากหลายของประชากร โดยสังเกตได้จากกราฟ 3.6 ซึ่งมาจากผลการทดลองในหัวข้อ 5.2.1 จะพบว่ากลายพันธุ์มีความหลากหลายของประชากรมากกว่าความหลากหลายของประชากรของ PSO ผลจากที่ประชากรมีความหลากหลายมาก ทำให้ประชากรกระจายตัวค้นหาในบริเวณกว้าง อนุภาคอาจจะค้นหาค้นหาขอบเขตอิทธิพลของจุดสุดสัมพัทธ์ซึ่งอาจจะพบตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์เดิม หลังจากนั้นอนุภาคทั้งหมดจะเคลื่อนไปที่ไปยังตำแหน่งใหม่ที่พบจึงสามารถหลุดออกจากจุดสุดสัมพัทธ์ได้ และจากกราฟ 3.7 และ 3.8 ซึ่งมาจากผลการทดลองในหัวข้อ 5.2.1 จะพบว่ากลายพันธุ์มีความเร็วของอนุภาคที่เปลี่ยนแปลงตลอดเวลาซึ่งแตกต่างจากความเร็วของอนุภาคของ PSO ที่มีค่าลู่เข้าหาศูนย์ หมายความว่า การกลายพันธุ์ทำให้อนุภาคเกิดการวิ่งกระจัดกระจายในบริเวณที่กว้างจึงทำให้การค้นหาเกิดขึ้นในบริเวณที่กว้าง ในขณะที่ PSO ทำให้อนุภาคเกิดการลู่เข้าหากันจึงทำให้การค้นหาเกิดขึ้นในบริเวณที่แคบ



รูปที่ 3.6 ความหลากหลายของ PSO และ NCM

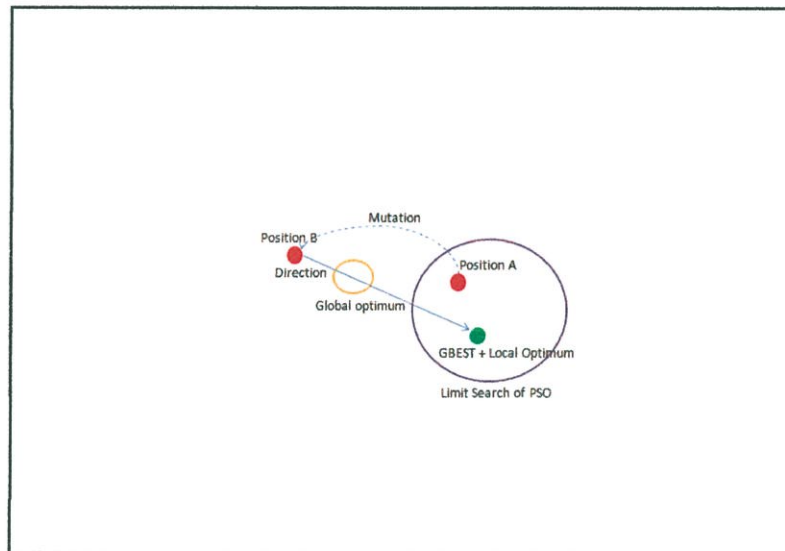


รูปที่ 3.7 ความเร็วของอนุภาคของ NCM



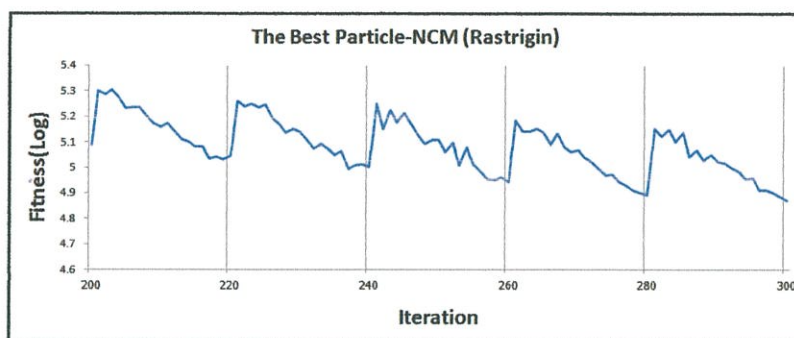
รูปที่ 3.8 ความเร็วของอนุภาคของ PSO

การทำงานของ PSO โดยปกติ อนุภาคทั้งหมดจะเคลื่อนที่เข้าหา GBEST โดยระหว่างทางถ้าพบตำแหน่งที่ดีกว่า GBEST ก็จะเอาตำแหน่งนั้นแทน GBEST และเมื่ออนุภาคจะเคลื่อนที่เข้าใกล้ GBEST อนุภาคจะค้นหาบริเวณรอบ GBEST เท่านั้น ถ้ากรณี GBEST อยู่ที่จุดสุดสัมพัทธ์จะส่งผลให้อนุภาคไม่สามารถค้นหาในบริเวณอื่นจึงไม่พบคำตอบที่ดีขึ้น การกลายพันธุ์วิธีนี้ช่วยปรับปรุง PSO โดยทำให้อนุภาคกระโดดออกจากบริเวณจุดสุดสัมพัทธ์ แล้วอนุภาคจะเคลื่อนกลับเข้ามายังจุดสุดสัมพัทธ์เดิม เนื่องจากตามหลักการการทำงานของ PSO อนุภาคจะต้องเคลื่อนที่ไล่เข้าหา GBEST แต่ในระหว่างการเคลื่อนที่กลับนั้นอนุภาคมีโอกาสพบตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ ทำให้สามารถปรับปรุงคำตอบได้ โดยสามารถแสดงดังรูป 3.9 และกราฟ 3.10 ซึ่งมาจากผลการทดลองในหัวข้อ 5.2.2



รูปที่ 3.9 ตัวอย่าง NCM ช่วยปรับปรุง PSO

จากรูป 3.9 GBEST (วงกลมสีเขียว) อยู่ภายในบริเวณตำแหน่งของจุดสุดสัมพัทธ์ (วงกลมสีม่วง) ทำให้อนุภาคทั้งหมดเคลื่อนที่เข้ามารวมกันภายในบริเวณตำแหน่งของจุดสุดสัมพัทธ์ และค้นหาบริเวณรอบจุดสุดสัมพัทธ์ ทำให้ไม่สามารถพบจุดสุดสัมบูรณ์ (วงกลมสีส้ม) เมื่อเกิดการกลายพันธุ์ทำให้อนุภาค (วงกลมสีแดง) กระโดดจากตำแหน่ง A ไปยังตำแหน่ง B หลังจากนั้นอนุภาคจะเคลื่อนที่กลับเข้ามายังจุดสุดสัมพัทธ์ ระหว่างทางมีโอกาสได้พบกับจุดสุดสัมบูรณ์

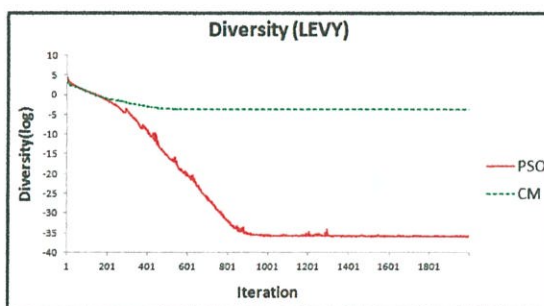


รูปที่ 3.10 ตัวอย่าง NCM ช่วยปรับปรุง PSO

จากรูป 3.10 การกลายพันธุ์ถูกกำหนดให้ดำเนินการทุกๆ 20 รอบของการค้นหา เมื่อผ่านการกลายพันธุ์จะพบว่าเส้นกราฟมีค่าเพิ่มขึ้น เพราะอนุภาคถูกกระจายตัวไปตกในตำแหน่งที่แย่กว่าเดิม หลังจากนั้นอนุภาคจะทำการลู่เข้าหากันใหม่ ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิม ทำให้ค่าตอบนั้นดีขึ้น สังเกตได้จากเส้นกราฟที่จะลู่ลงไปมากกว่าเดิมก่อนการกลายพันธุ์ในครั้งก่อน

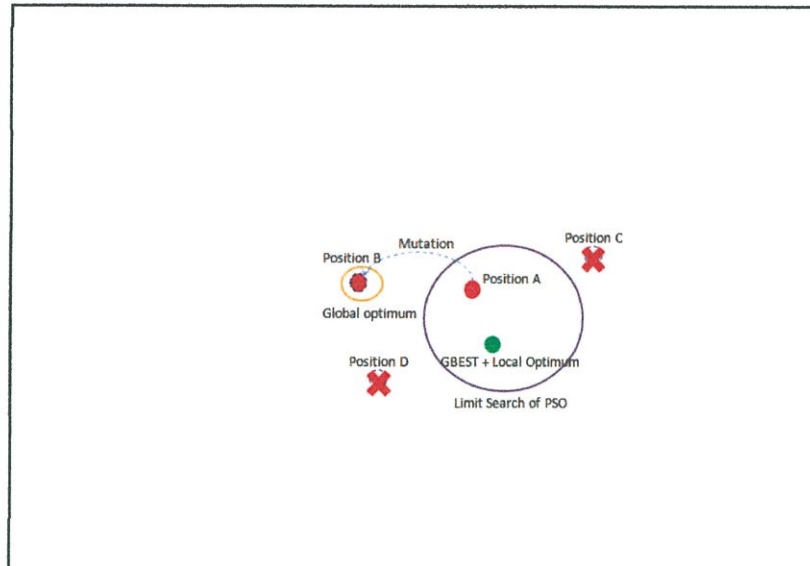
3.1.6 ข้อดีของการกลายพันธุ์ประเภทมีการเปรียบเทียบ

เช่นเดียวกับการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ การกลายพันธุ์ประเภทนี้ได้เพิ่มความหลากหลายให้กับประชากร จึงสามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ และได้รับผลลัพธ์การค้นหาที่ดีขึ้น โดยสังเกตได้จากกราฟ 3.11 ซึ่งมาจากผลการทดลองในหัวข้อ 5.2.3 จะพบว่ากลายพันธุ์ประเภทนี้มีความหลากหลายของประชากรมากกว่าความหลากหลายของประชากรของ PSO



รูปที่ 3.11 ความหลากหลายของ PSO และ CM

เมื่ออนุภาคติดอยู่ในจุดสุดสัมพัทธ์ การกลายพันธุ์ช่วยปรับปรุง PSO โดยจะทำการสุ่มตำแหน่งรอบจุดสุดสัมพัทธ์ในบริเวณที่กว้างกว่าขอบเขตการค้นหาของ PSO ส่งผลให้ความหลากหลายของประชากรมีค่าเพิ่มขึ้น โดยสังเกตจากกราฟ 3.11 พบว่าการกลายพันธุ์ทำให้ความหลากหลายของประชากรมีค่ามากกว่าความหลากหลายของประชากรของ PSO ดังนั้นขอบเขตในการค้นหา ณ ช่วงเวลาที่ติดอยู่ในจุดสุดสัมพัทธ์จึงมากกว่า PSO เมื่อการกลายพันธุ์สุ่มไปพบตำแหน่งที่ดีขึ้นกว่าจุดสุดสัมพัทธ์เดิมก็จะทำให้เกิด GBEST ใหม่ หลังจากนั้นอนุภาคทั้งหมดจะเคลื่อนที่ออกจากจุดสุดสัมพัทธ์เดิมและไปค้นหาในบริเวณ GBEST ใหม่โดยสามารถแสดงดังรูปที่ 3.12



รูปที่ 3.12 ตัวอย่าง CM ช่วยปรับปรุง PSO

จากรูป 3.12 GBEST (วงกลมสีเขียว) อยู่ภายในบริเวณตำแหน่งของจุดสุดสัมพัทธ์ (วงกลมสีม่วง) ทำให้อนุภาคทั้งหมดเคลื่อนที่เข้ามารวมกันภายในบริเวณตำแหน่งของจุดสุดสัมพัทธ์ และค้นหาบริเวณรอบจุดสุดสัมพัทธ์ จึงไม่สามารถพบจุดสุดสัมบูรณ์ เมื่อเกิดการกลายพันธุ์จะทำการสุ่มตำแหน่งไปรอบวงกลมสีแดง ซึ่งจะมีขอบเขตการค้นหามากกว่า PSO ในกรณีตำแหน่งสุ่มแยกกว่าตำแหน่งของอนุภาค เช่น ตำแหน่ง C และ D ตำแหน่งที่สุ่มนั้นจะไม่นำมาแทนที่ตำแหน่งของอนุภาค แต่ในกรณีตำแหน่งที่สุ่มดีกว่าตำแหน่งของอนุภาค เช่น ตำแหน่ง B ตำแหน่งที่สุ่มจะนำมาแทนที่ตำแหน่งของอนุภาคและผลจากการสุ่มรอบจุดสุดสัมพัทธ์ทำให้มีโอกาสพบจุดสุดสัมบูรณ์

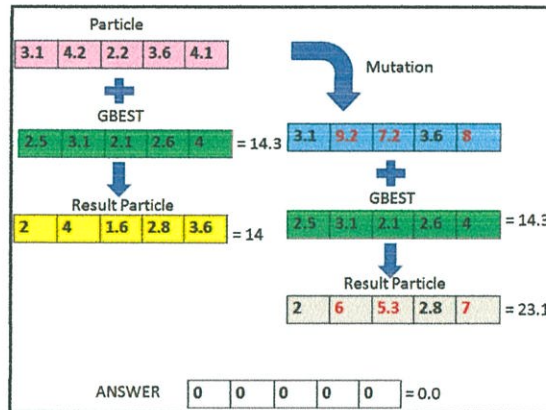
3.1.7 ข้อเสียของการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ

การกลายพันธุ์ประเภทไม่มีการเปรียบเทียบช่วยลดปัญหาการติดในจุดสุดสัมพัทธ์ แต่ผลของการกลายพันธุ์ทำให้เกิดปัญหาใหม่ คือปัญหาความยุ่งยากในการปรับพารามิเตอร์ของการกลายพันธุ์ให้เหมาะสมกับปัญหาที่แก้ไข จากความยุ่งยากนี้ทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ (Obstruction mutation Problem) [84] และปัญหาการกลายพันธุ์น้อยเกินไป (Under-mutated Problem) [84] โดยรายละเอียดของปัญหาทั้งสองมีดังนี้

3.1.7.1 ปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

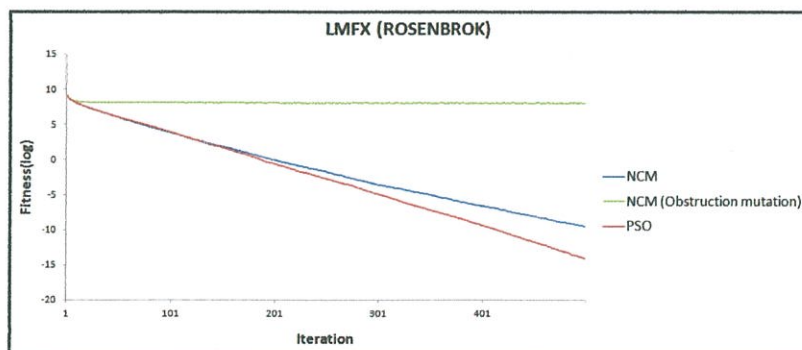
โดยปกติการทำงานของ PSO อนุภาคทั้งหมดเคลื่อนที่เข้าหา GBEST โดยจะใช้ระยะเวลาช่วงหนึ่งเพื่อให้อนุภาคเข้าใกล้กัน ซึ่งเหตุการณ์นี้เรียกว่าการลู่เข้าหากันหลังจากนั้น ก็ จะทำการค้นหาในบริเวณรอบ GBEST แต่ระหว่างการลู่เข้าหากันถ้าหากเกิดการกลายพันธุ์ขึ้น การกลายพันธุ์จะทำให้อนุภาคกระจายตัวออกจากกันแบบสุ่ม หลังจากการกลายพันธุ์ อนุภาคลู่เข้าหา GBEST อีกครั้ง แต่ถ้าหากการกลายพันธุ์เกิดขึ้นระหว่างการลู่เข้าหากันอีกครั้ง อนุภาคกระจายตัวออกจากกันแบบสุ่มอีกครั้ง และไม่สามารถเข้าไปค้นหารอบ GBEST ได้ ดังนั้นการกลายพันธุ์จึงขัดขวางการลู่เข้าหากันของอนุภาค ทำให้ผลลัพธ์การค้นหาเสมือนกับการค้นหาแบบสุ่ม (Random

Search) ซึ่งเหตุการณ์นี้เกิดจากการกำหนดพารามิเตอร์ให้ดำเนินของการกลายพันธุ์มากเกินไป จนทำให้อนุภาคไม่สามารถลู่เข้าหากันแต่กลับกระจายตัวออกจากกันไปเรื่อยๆ เรียกปัญหานี้ว่า ปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ ผลกระทบจากปัญหานี้ คือคำตอบที่ได้จากการค้นหาจะแยกว่าไม่ดำเนินการกลายพันธุ์ โดยปัญหานี้สามารถสาธิตให้อยู่ในรูป 3.13 และกราฟ 3.14 ซึ่งมาจากผลการทดลองในหัวข้อ 5.3.1 นอกจากนี้จากตารางผลการทดลอง 5.7 ซึ่งมาจากผลการทดลองในหัวข้อ 5.3.1 แสดงให้เห็นว่าถ้าหากกำหนดพารามิเตอร์ให้ดำเนินการกลายพันธุ์มากเกินไป ผลลัพธ์จากการค้นหาโดยรวมแล้วจะแยกว่า PSO ที่ไม่ดำเนินการกลายพันธุ์



รูปที่ 3.13 ตัวอย่างปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

จากรูป 3.13 คำตอบของปัญหานี้ คือทุกมิติต้องมีค่าเป็น 0 การค้นหาคำตอบด้วย PSO ซึ่งทำหน้าที่ค้นหาค่าเพื่อนำมาแทนมิติต่างๆ ให้ได้ค่าความเหมาะสมเท่ากับ 0 หรือทุกมิติเป็น 0 ในสถานการณ์นี้ มีอนุภาคหนึ่งอนุภาค (ช่องตารางสีเขียว) ได้รับอิทธิพลจาก GBEST (ช่องตารางสีเขียวย) จะทำให้ได้ผลลัพธ์เป็นอนุภาคใหม่ (ช่องตารางสีเหลือง) ค่าความเหมาะสมที่ได้คือ 14 ซึ่งทำให้อนุภาคนั้นเคลื่อนที่เข้าหาคำตอบมากขึ้น แต่ถ้าหากดำเนินการกลายพันธุ์ ซึ่งการกลายพันธุ์จะเข้ามาแทรกทำให้อนุภาคนั้นเปลี่ยนไปจากเดิม (ช่องตารางสีฟ้า) เมื่อได้รับอิทธิพลจาก GBEST จะทำให้ได้ผลลัพธ์เป็นอนุภาคใหม่ (ช่องตารางสีเทา) ค่าความเหมาะสมที่ได้นั้นคือ 23.1 ซึ่งได้ผลลัพธ์ที่แยกว่าไม่ดำเนินการกลายพันธุ์ ถ้าเหตุการณ์นี้เกิดขึ้นทุกๆ ครั้งที่ดำเนินการค้นหา และเกิดขึ้นทุกๆ อนุภาค ค่าความเหมาะสมที่ได้นั้น จะไม่มีทางลดลงไปเป็น 0 ได้เลย เนื่องจากอนุภาคต่างๆ ถูกขัดขวางด้วยการกลายพันธุ์

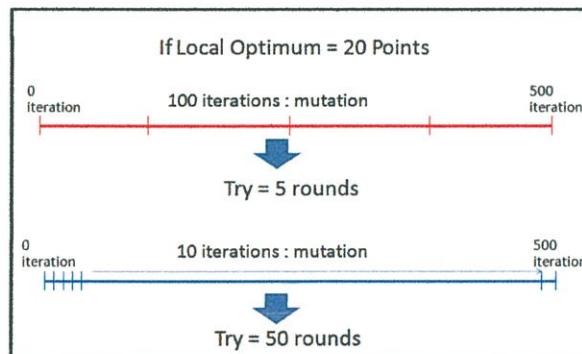


รูปที่ 3.14 กราฟแสดงตัวอย่างปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

จากรูป 3.14 เส้นสีเขียวแทนการกลายพันธุ์มากเกินไป ซึ่งทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาค่าตอบ เส้นสีน้ำเงินแทนการกลายพันธุ์ที่เหมาะสม และเส้นสีแดงแทน PSO จากกราฟจะเห็นว่า ถ้ากำหนดให้ดำเนินการการกลายพันธุ์มากเกินไปจะทำให้อนุภาคไม่คู่เข้าหากัน โดยจากเส้นกราฟจะมีค่าคงที่และไม่มีแนวโน้มที่จะลดลงมา เมื่อเปรียบเทียบกับการกลายพันธุ์ที่ดำเนินการเหมาะสมและ PSO เส้นกราฟจะมีค่าลดลงไปเรื่อยๆ เพราะอนุภาคสามารถคู่เข้าหากันได้ จึงได้รับคำตอบที่ได้จึงดีกว่า

3.1.7.2 ปัญหาการกลายพันธุ์น้อยเกินไป

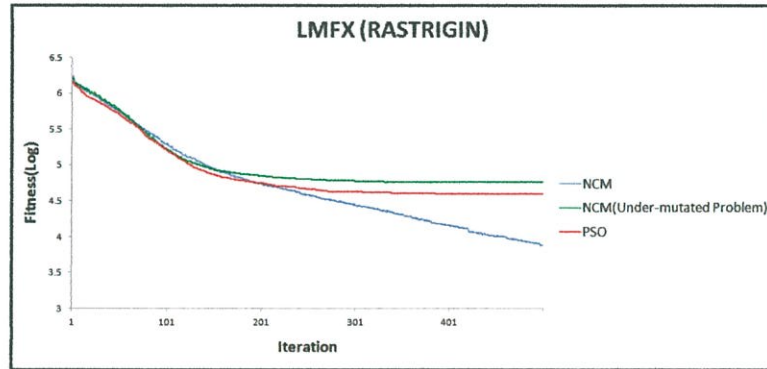
ในช่วงเวลาที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ คำตอบที่ได้จากการค้นหาจะไม่ดีขึ้น เนื่องจากอนุภาคทั้งหมดจะค้นหาในบริเวณรอบจุดสุดสัมพัทธ์ ถ้าในช่วงเวลานี้ดำเนินการกลายพันธุ์ก็จะช่วยเพิ่มการกระจายตัวให้กับอนุภาค ทำให้มีโอกาสออกไปค้นหาในบริเวณอื่นและมีโอกาสพบคำตอบที่ดีขึ้น แต่ถ้าในช่วงเวลานี้ไม่ดำเนินการกลายพันธุ์ปล่อยให้ค้นหาต่อไปเป็นเวลานานๆ แล้วจึงเริ่มดำเนินการกลายพันธุ์ ทำให้สูญเสียโอกาสในการกระจายตัวออกไปค้นหา ซึ่งเหตุการณ์นี้เกิดจากการกำหนดพารามิเตอร์ให้ดำเนินการของการกลายพันธุ์น้อยเกินไป จึงไม่ส่งผลให้เกิดความหลากหลายที่มากพอที่จะแก้ไขปัญหาการติดในจุดสุดสัมพัทธ์ เทคนิคการกลายพันธุ์เพิ่มเข้าไปจึงไม่มีประโยชน์ เรียกปัญหานี้ว่าปัญหาการกลายพันธุ์น้อยเกินไป ผลกระทบสำหรับปัญหานี้คือคำตอบที่ได้นั้นจะไม่ดีขึ้นเมื่อเปรียบเทียบกับ PSO ที่ไม่ดำเนินการกลายพันธุ์ โดยปัญหานี้สามารถสาธิตให้อยู่ในรูป 3.15 และกราฟ 3.16 ซึ่งมาจากผลการทดลองในหัวข้อ 5.3.2 นอกจากนี้จากตารางผลการทดลอง 5.8 ซึ่งมาจากผลการทดลองในหัวข้อ 5.3.2 แสดงให้เห็นว่าถ้าหากกำหนดพารามิเตอร์ให้ดำเนินการกลายพันธุ์น้อยเกินไปคำตอบที่ได้รับโดยรวมแล้วจะใกล้เคียงกับ PSO ที่ไม่ดำเนินการกลายพันธุ์



รูปที่ 3.15 ตัวอย่างปัญหาการกลายพันธุ์น้อยเกินไปของ NCM

จากรูป 3.15 สมมติสถานการณ์นี้ไม่เกิดปัญหาการกลายพันธุ์มากเกินไป โดยปัญหานี้มีจุดสุดสัมพัทธ์อยู่ 20 จุด ทำการค้นหาสองแบบ แบบแรกดำเนินการกลายพันธุ์เมื่อรอบการค้นหาคำตอบสามารถหารด้วย 100 ลงตัว (เส้นสีแดง) แบบสองดำเนินการกลายพันธุ์เมื่อรอบการค้นหาคำตอบสามารถหารด้วย 10 ลงตัว (เส้นสีน้ำเงิน) ถ้ารอบการค้นหาทั้งหมดมี 1000 รอบ แบบแรกจะมีโอกาสในการกระโดดออกจากจุดสุดสัมพัทธ์เพียง 5 ครั้งเท่านั้น ในขณะที่แบบที่สองจะมีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ 50 ครั้ง นั่นหมายความว่าแบบที่สองจะมีโอกาสในการกระโดดออกจากจุดสุดสัมพัทธ์มากกว่าแบบแรกถึงประมาณ 10 เท่า ถ้ามีจุดสุดสัมพัทธ์ 20 จุด โอกาสที่แบบ

แรกจะหาคำตอบพบนั้นแทบจะไม่มีเลย เพราะสามารถหลีกเลี่ยงจุดสุดสัมพัทธ์มากที่สุดได้เพียงแค่ 5 จุด ในขณะที่แบบที่สองมีโอกาสที่จะหลีกเลี่ยงจุดสุดสัมพัทธ์ได้หมด ทำให้ประสิทธิภาพของแบบแรกด้อยกว่าแบบที่สองมากทั้งๆ ที่ใช้จำนวนรอบในการค้นหาเท่ากัน ในกรณีนี้แบบแรกคือตัวแทนของปัญหาหลายพันธุ่น้อยเกินไป



รูปที่ 3.16 ตัวอย่างปัญหาหลายพันธุ่น้อยเกินไปของ NCM

จากรูป 3.16 เส้นสีเขียวแทนการกลายพันธุ่น้อยเกินไป เส้นสีน้ำเงินแทนการกลายพันธุที่เหมาะสม และเส้นสีแดงแทน PSO จากกราฟจะเห็นว่าถ้ากำหนดให้ดำเนินการการกลายพันธุ่น้อยเกินไป ผลลัพธ์จากการค้นหาจะเหมือนกับ PSO โดยเส้นกราฟของการกลายพันธุ่น้อยเกินไปและ PSO จะมีค่าคงที่เหมือนกันและไม่มีแนวโน้มที่จะดีขึ้น เมื่อเปรียบเทียบกับการกลายพันธุที่เหมาะสมเส้นกราฟจะมีค่าดีขึ้นเรื่อยๆ จึงได้รับคำตอบที่ได้จึงดีกว่า

3.1.8 ข้อเสียของการกลายพันธุประเภทมีการเปรียบเทียบ

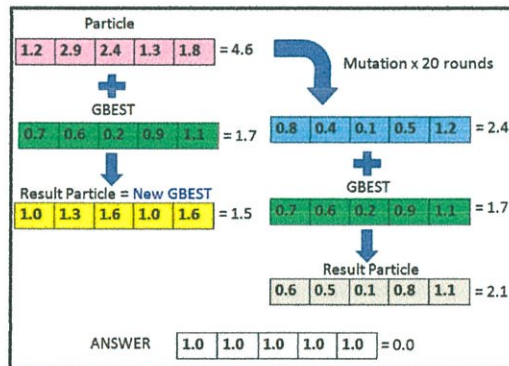
เช่นเดียวกับการกลายพันธุประเภทไม่มีการเปรียบเทียบ การกลายพันธุประเภทนี้สามารถลดปัญหาการติดในจุดสุดสัมพัทธ์ แต่ผลการกลายพันธุทำให้เกิดปัญหาใหม่ คือปัญหาความยุ่งยากในการปรับพารามิเตอร์ของการกลายพันธุให้เหมาะสมกับปัญหาที่แก้ไข จากความยุ่งยากนี้ทำให้เกิดปัญหาการลู่เร็วเกินไป (Premature Convergence Problem) [84] และปัญหาหลายพันธุ่น้อยเกินไป (Under-mutated Problem) [84] โดยรายละเอียดของปัญหามีดังนี้

3.1.8.1 ปัญหาการลู่เร็วเกินไป

ถ้าสมมติให้จำนวนของอนุภาคของ PSO เท่ากับจำนวนอนุภาคของ CM แล้วกำหนดให้ CM นั้นดำเนินการกลายพันธุมากๆ CM จะมีผลลัพธ์ที่ดีกว่า PSO เพราะ CM มีจำนวนการเดาคำตอบมากกว่า PSO เนื่องจาก CM มีผลจากการค้นหาของการกลายพันธุรวมกับผลการค้นหาของ PSO จึงมีจำนวนการเดาคำตอบมากกว่า PSO ตัวอย่างเช่น กำหนดให้ PSO มีจำนวนอนุภาคเท่ากับ 20 อนุภาค และ CM มีจำนวนอนุภาคเท่ากับ 20 อนุภาคเช่นกันแต่กำหนดให้ดำเนินการกลายพันธุต่อ 1 อนุภาคและต่อ 1 รอบการค้นหาคำตอบเท่ากับ 10 ครั้ง ในการเดาคำตอบหนึ่งรอบการค้นหาคำตอบของ PSO คือ 20 แต่ในการเดาคำตอบหนึ่งรอบการค้นหาคำตอบของ CM คือ 220 ซึ่งหมายความว่า CM มีการเดาคำตอบที่มากกว่า PSO ประมาณ 10 เท่า ซึ่งถึงจะไม่ยุติธรรมและแน่นอนผลลัพธ์การค้นหาที่ดีกว่า PSO เพื่อให้เห็นปัญหาการลู่เร็วเกินไปที่เกิดขึ้นกับ CM ต้อง

เปรียบเทียบกับ PSO โดยมีจำนวนการเดาคำตอบที่เท่ากัน ตัวอย่างเช่น กำหนดให้ PSO มีจำนวนอนุภาคเท่ากับ 220 อนุภาค เปรียบเทียบกับ CM มีจำนวนอนุภาคเท่ากับ 20 อนุภาค และกำหนดให้ดำเนินการกลายพันธุ์ต่อ 1 อนุภาคและต่อ 1 รอบการค้นหาคำตอบเท่ากับ 10 รอบ ซึ่งในกรณีนี้ CM มีโอกาสได้ผลลัพธ์การค้นหาที่แยกว่า PSO

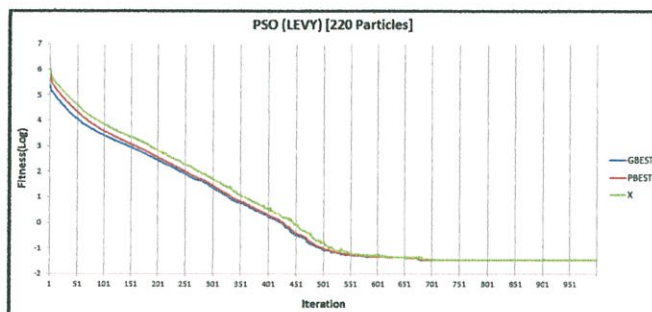
โดยปกติ PSO ประสบปัญหาการลู่เร็วเกินไปอยู่แล้ว ผลเนื่องมาจากการที่อนุภาคทั้งหมดนั้นลู่เข้าหา GBEST ถ้าการกลายพันธุ์ช่วยให้เกิดตำแหน่งใหม่แทนที่ตำแหน่งเดิม โดยตำแหน่งใหม่เป็นตำแหน่งที่ดีกว่าเดิมเสมอ มันส่งผลให้อนุภาคเคลื่อนที่เข้าหา GBEST เร็วขึ้นกว่าเดิม ผลจากการที่อนุภาคเคลื่อนที่เข้าหา GBEST เร็วเกินไป จะทำให้ขอบเขตของการค้นหาจะแคบลงอย่างรวดเร็ว ซึ่งส่งผลให้ปัญหาการลู่เร็วเกินไปนั้นรุนแรงขึ้นกว่าเมื่อเปรียบเทียบกับ PSO ที่ไม่ดำเนินการกลายพันธุ์และมีจำนวนการเดาคำตอบที่เท่ากัน เพราะ PSO จะค่อยๆ ลู่เข้าหากัน จึงทำให้ขอบเขตการค้นหาค่อยๆ ลดลง โดยเหตุการณ์นี้เกิดจากการกำหนดพารามิเตอร์ให้ดำเนินการของการกลายพันธุ์มากเกินไป จึงส่งผลให้อนุภาคลู่เข้าหา GBEST เร็วกว่า PSO ปกติมาก จึงเกิดปัญหาการลู่เร็วเกินไปมากขึ้น ซึ่งจะทำให้บางปัญหานั้นติดในจุดสุดสัมพัทธ์ได้ง่ายขึ้น ผลกระทบสำหรับปัญหานี้ คือคำตอบที่ได้จากการค้นหาจะแยกว่า PSO ไม่ดำเนินการกลายพันธุ์และมีจำนวนการเดาคำตอบที่เท่ากัน โดยปัญหานี้สามารถสาธิตให้อยู่ในรูป 3.17 และกราฟ 3.18 และ 3.19 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 นอกจากนี้จากตารางผลการทดลอง 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แสดงให้เห็นว่า ในบางปัญหาเมื่อนำ CM ไปประยุกต์ใช้ โดยกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป จะได้ผลลัพธ์ที่แยกว่า PSO ไม่ดำเนินการกลายพันธุ์และมีจำนวนการเดาคำตอบที่เท่ากัน



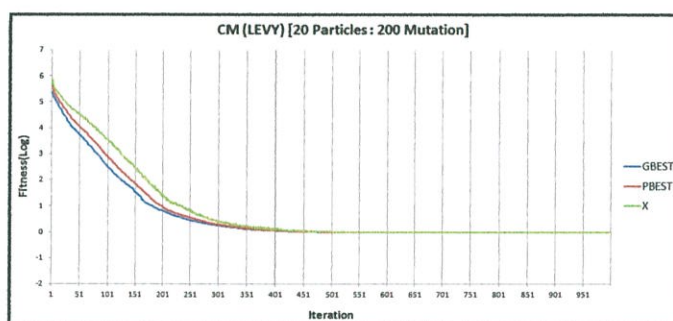
รูปที่ 3.17 ตัวอย่างปัญหาการลู่เร็วเกินไปของ CM

จากรูป 3.17 คำตอบของปัญหานี้ คือทุกมิติต้องมีค่าเป็น 1 การค้นหาคำตอบด้วย PSO ซึ่งทำหน้าที่ค้นหาค่าเพื่อนำมาแทนมิติต่างๆ ให้ได้ค่าความเหมาะสมเท่ากับ 0 หรือทุกมิติเป็น 1 ในสถานการณ์นี้ มีอนุภาคหนึ่งอนุภาค (ช่องตารางสีเขียว) ได้รับอิทธิพลจาก GBEST (ช่องตารางสีเขียวย) จะทำให้ได้ผลลัพธ์เป็นอนุภาคใหม่ (ช่องตารางสีเหลือง) ค่าความเหมาะสมที่ได้คือ 1.5 ซึ่งเป็นค่า GBEST ใหม่ ที่เกิดขึ้นจากการคำนวณของ PSO แต่ถ้าเกิดการกลายพันธุ์จำนวน 20 รอบ แทรกขึ้นทำให้อนุภาคนั้นเปลี่ยนไปจากเดิม (ช่องตารางสีฟ้า) เมื่อได้รับอิทธิพลจาก GBEST จะทำให้ได้ผลลัพธ์เป็นอนุภาคใหม่ (ช่องตารางสีเทา) ค่าความเหมาะสมที่ได้นั้นคือ 2.1 ซึ่งได้ผลลัพธ์ที่แยกว่าไม่ดำเนินการกลายพันธุ์ เนื่องจากอนุภาคได้ผลจากการกลายพันธุ์ทำให้อนุภาคเข้ามารวมตัวกันอย่างรวดเร็วโดยการเปลี่ยนแปลงตัวเองให้ใกล้เคียงกับ GBEST มากเกินไป ซึ่งเป็นการลดโอกาสการ

เคลื่อนที่ของอนุภาคที่ค่อยๆ เคลื่อนตัวเข้าหา GBEST แล้วพบคำตอบในระหว่างทางที่ลู่เข้ากัน ตามหลักของ PSO ปกติ



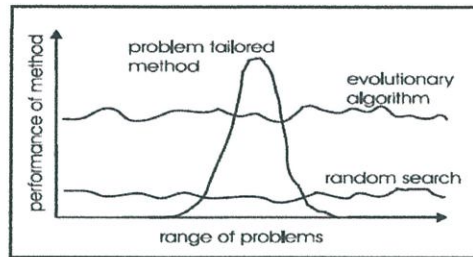
รูปที่ 3.18 การลู่เข้าหากันของ PSO



รูปที่ 3.19 การลู่เข้าหากันของ CM

จากรูป 3.18 กราฟแสดงค่าลอการิทึมเฉลี่ยของ GBEST PBEST และตำแหน่งของอนุภาค (X) ของ PSO ที่มีจำนวนอนุภาคเท่ากับ 220 อนุภาค และสำหรับรูป 3.19 กราฟแสดงค่าลอการิทึมเฉลี่ยของของ GBEST PBEST และตำแหน่งของอนุภาค (X) ของ CM ที่มีจำนวนอนุภาคเท่ากับ 20 อนุภาคและมีจำนวนในการกลายพันธุ์ต่ออนุภาคต่อรอบการค้นหาคำตอบคือ 10 ครั้ง สังเกตเส้นกราฟของ CM จะลู่เข้าสู่สถานะคงที่ก่อน PSO ปกติหมายความว่า CM มีความเร็วในการลู่เข้าหากันมากกว่า PSO ซึ่งทำให้โอกาสการติดจุดสุดสัมพัทธ์นั้นเกิดขึ้นได้ง่ายกว่า PSO ดังนั้นการกลายพันธุ์ช่วยเร่งให้อนุภาคลู่เข้าหากันเร็วขึ้น ซึ่งเป็นเหตุให้เกิดปัญหาการลู่เร็วเกินไปมากกว่า PSO

อย่างไรก็ตามปัญหาการลู่เร็วเกินไปที่เกิดจาก CM เมื่อเปรียบเทียบกับปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบที่เกิดจาก NCM ปัญหาการลู่เร็วเกินไปมีผลกระทบต่อการค้นหาคำตอบน้อยกว่ามาก เนื่องจาก CM ใช้ผลลัพธ์จากการกลายพันธุ์เฉพาะที่มีตำแหน่งดีกว่าตำแหน่งเดิมเท่านั้น ซึ่งทำให้อนุภาคเข้าหา GBEST เร็วขึ้น จึงสูญเสียการค้นหาระหว่างที่กำลังลู่เข้าหา GBEST แต่ทิศทางของอนุภาคทั้งหมดยังคงลู่เข้าหา GBEST เช่นเดียวกับ PSO หลักการทำงานของ PSO ยังคงมีประสิทธิภาพอยู่ ในขณะที่ปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบที่เกิดกับ NCM ทำให้การค้นหาไม่เป็นไปตามหลักการทำงานของ PSO เพราะอนุภาคกระจายตัวแบบสุ่ม จึงกลายเป็นการค้นหาแบบสุ่มผลการค้นหาจึงแยกว่า PSO

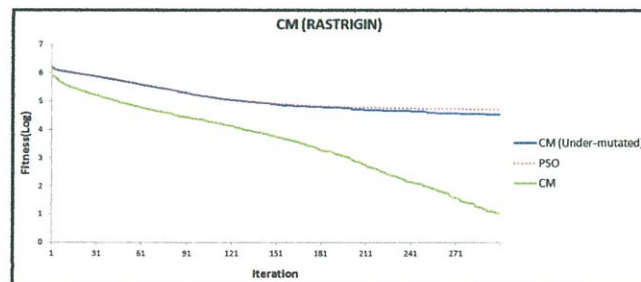


รูปที่ 3.20 ประสิทธิภาพการทำงานของขั้นตอนวิธีวิวัฒนาการ [71]

จากกราฟ 3.20 ซึ่งมาจากเอกสาร [71] แสดงให้เห็นว่าวิธีแก้ปัญห เฉพาะเจาะจงสามารถแก้ปัญหาใดปัญหาหนึ่งได้อย่างมีประสิทธิภาพเท่านั้น แต่ EA หรือ PSO สามารถแก้ปัญหาโดยเฉลี่ยได้อย่างมีประสิทธิภาพทั้งหมด และดีกว่าวิธีการค้นหาคำตอบแบบสุ่ม ทั้งหมด ดังนั้นผลกระทบจากการปรับพารามิเตอร์ผิดพลาดที่เกิดกับ CM จึงน้อยกว่า NCM มาก ซึ่ง การกำหนดพารามิเตอร์ NCM ให้เหมาะสมกับการแก้ปัญหาทั่วไปนั้นทำได้ยาก เพราะถ้ากำหนด พารามิเตอร์ไม่เหมาะสมกับปัญหาที่แก้ไข ผลลัพธ์ที่ได้จากการค้นหาจะแย่กว่า PSO ที่ไม่ดำเนินการ กลายพันธุ์มาก ในขณะที่การกำหนดพารามิเตอร์ CM ให้เหมาะสมกับการแก้ปัญหาทั่วไปนั้นทำได้ง่าย กว่า NCM เพราะถ้ากำหนดพารามิเตอร์ไม่เหมาะสมกับปัญหาที่แก้ไข ผลลัพธ์ที่ได้จากการค้นหาจะแย่ กว่า PSO ที่ไม่ดำเนินการกลายพันธุ์เพียงเล็กน้อย เนื่องจากผลกระทบจากการกลายพันธุ์ที่น้อยกว่า จึงทำให้การประยุกต์ใช้ CM กับปัญหาทั่วไปทำได้ง่ายกว่า ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้เลือกวิธี CM มาปรับปรุงให้มีประสิทธิภาพดีขึ้น

3.1.8.2 ปัญหาการกลายพันธุ์น้อยเกินไป

สำหรับปัญหาการกลายพันธุ์น้อยเกินไปของ CM มีลักษณะเช่นเดียวกับ NCM คือการกลายพันธุ์ถูกกำหนดให้ดำเนินการน้อยเกินไป จึงไม่สามารถช่วยให้กระจายออกจากขอบเขต ของจุดสุดสัมพัทธ์ และไม่ส่งผลให้เกิดความหลากหลายที่มากพอแก้ไขปัญหาคัดในจุดสุดสัมพัทธ์ เทคนิคการกลายพันธุ์ที่เพิ่มเข้าไปจึงไม่มีประโยชน์ ผลกระทบสำหรับปัญหานี้คือคำตอบที่ได้นั้นไม่มี ประสิทธิภาพเท่าที่ควร โดยปัญหานี้สามารถแสดงให้เห็นดังกราฟ 3.21 ซึ่งมาจากผลการทดลองใน หัวข้อ 5.4.3 นอกจากนี้จากตารางผลการทดลอง 5.10 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.3 แสดง ให้เห็นว่าถ้าหากกำหนดพารามิเตอร์ให้ดำเนินการกลายพันธุ์น้อยเกินไปคำตอบที่ได้รับโดยรวมแล้วจะ ใกล้เคียงกับ PSO ที่ไม่ดำเนินการกลายพันธุ์



รูปที่ 3.21 ตัวอย่างปัญหาการกลายพันธุ์น้อยเกินไปของ CM

จากรูป 3.21 เส้นสีเขียวแทนการกลายพันธุ์ที่เหมาะสม เส้นสีน้ำเงินแทนการกลายพันธุ์น้อยเกินไป และเส้นสีแดงแทน PSO จากกราฟจะเห็นว่า ถ้ากำหนดให้ดำเนินการการกลายพันธุ์น้อยเกินไป ผลลัพธ์จากการค้นหาจะเหมือนกับ PSO โดยเส้นกราฟของการกลายพันธุ์น้อยเกินไปและ PSO จะมีค่าคงที่และไม่มีแนวโน้มที่จะดีขึ้นเหมือนกัน เมื่อเปรียบเทียบกับกรการกลายพันธุ์ที่ดำเนินการเหมาะสมเส้นกราฟจะมีค่าดีขึ้นเรื่อยๆ จึงได้รับคำตอบที่ได้จึงดีกว่า

3.2 การประยุกต์วิธีการเริ่มต้นการค้นหาใหม่กับ PSO

ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO สามารถแก้ไขด้วยการนำการกลายพันธุ์ มาประยุกต์ใช้ แต่อย่างไรก็ตามวิธีการแก้ไขปัญหาการติดในจุดสุดสัมพัทธ์ยังมีวิธีหนึ่งที่นิยม คือวิธีการเริ่มต้นการค้นหาใหม่หรือรีรัน (RERUN) [48-52] ซึ่งวิธีนี้สามารถนำมาประยุกต์ใช้ได้ง่ายเช่นเดียวกับการกลายพันธุ์ แต่ให้ผลกระทบจากการแก้ปัญหการติดในจุดสุดสัมพัทธ์ที่แตกต่างกัน โดยจากงานวิจัย [48-52] แสดงให้เห็นว่ารีรันสามารถเพิ่มประสิทธิภาพของ PSO

3.2.1 นิยามของการเริ่มต้นการค้นหาใหม่

การเริ่มต้นการค้นหาใหม่หรือรีรัน (RERUN) [48-52] คือการกระจายอนุภาคโดยตำแหน่งของอนุภาคที่ถูกรีเซ็ตจะถูกกำหนดค่าเริ่มต้นให้ใหม่ ซึ่งวิธีนี้ตำแหน่งการค้นหาก่อนการรีเซ็ตจะไม่ถูกนำมาใช้

3.2.2 นิยามของการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม

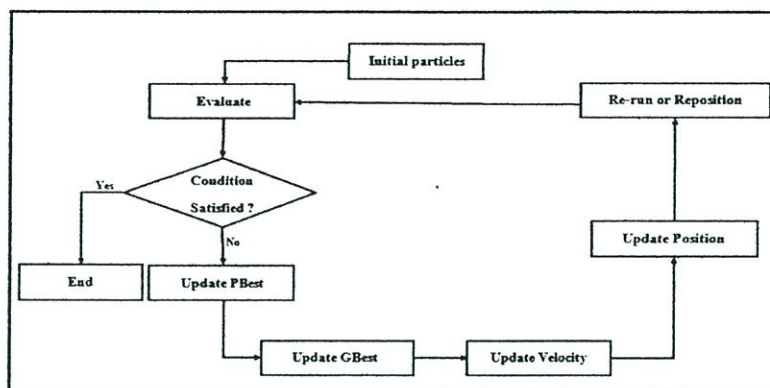
การเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิมหรือรีโพซิชัน (REPOSITION) [84] คือการกระจายอนุภาคโดยตำแหน่งของอนุภาคที่ถูกรีเซ็ตจะถูกดำเนินการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ (NCM) ในจำนวนมาก เช่น 30%, 50%, 70%, 90% ของจำนวนมิติทั้งหมดของทุกอนุภาค เป็นต้น โดยจะเรียกค่านี้ว่าความน่าจะเป็นในการรีโพซิชัน (Reposition Probability) หรือความน่าจะเป็นในการรีเซ็ต (Reset Probability) ซึ่งค่านี้จะถูกกำหนดให้และเป็นค่าคงที่ เพื่อให้อนุภาคกระจายตัวออกเป็นบริเวณกว้างมากพอที่ให้อนุภาคค้นหาออกจากบริเวณจุดสุดสัมพัทธ์ที่ติดอยู่ ซึ่งวิธีนี้ตำแหน่งการค้นหาก่อนการรีเซ็ตจะมีอิทธิพลต่อตำแหน่งหลังจากการรีเซ็ต

สำหรับหลักการทำงานของรีรันและรีโพซิชันมีหลักการทำงานที่เหมือนกันดังนี้ เมื่อ PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ เสมือนกับการค้นหาของ PSO ได้สิ้นสุดลง แต่เนื่องจากปัญหาการติดในจุดสุดสัมพัทธ์นี้สามารถตรวจสอบการเกิดขึ้นได้ง่าย เพราะเมื่อเกิดปัญหานี้ขึ้น GBEST จะไม่มีค่าดีขึ้นเป็นระยะเวลาานาน เพื่อแก้ปัญหานี้จึงนำรีรันหรือรีโพซิชันมาประยุกต์ใช้กับ PSO โดยหลักการคือเมื่อตรวจพบที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้นก็จะดำเนินขั้นตอนการรีรันหรือรีโพซิชัน หลังจากนั้นก็จะดำเนินการค้นหาแบบ PSO จนพบว่าเกิดปัญหาการติดในจุดสุดสัมพัทธ์อีกครั้ง จึงเริ่มขั้นตอนการรีรันหรือรีโพซิชันใหม่อีกครั้งวนเช่นนี้ไปเรื่อยๆ จนกว่าจะครบรอบการค้นหาหรือพบคำตอบ วิธีการนี้จะทำให้อนุภาคมีโอกาสกระจายตัวไปค้นหาในบริเวณอื่น ซึ่งอาจจะพบคำตอบที่ดีกว่าเดิม

โดยรีรันและรีโพซิชันนี้สามารถที่จะรีเซ็ตอนุภาคใหม่ทั้งหมดหรือรีเซ็ตอนุภาคใหม่เพียงบางส่วน สำหรับการรีเซ็ตเพียงบางส่วนในบางงานวิจัยเรียกว่าการกลายพันธุ์ [48, 49, 51]

นอกจากการกำหนดให้รีเซ็ตตำแหน่งของอนุภาคแล้ว รีรันและรีโพซิชันสามารถที่จะกำหนดให้รีเซ็ตส่วนประกอบของ PSO อื่นๆ เช่น PBEST GBEST ความเร็วของอนุภาค ซึ่งการกำหนดรีเซ็ตค่าเหล่านี้ขึ้นอยู่กับผู้ออกแบบขั้นตอนวิธีและผลของการรีเซ็ตค่าเหล่านี้จะส่งผลกระทบต่อการค้นหาที่แตกต่างกัน

3.2.3 ขั้นตอนของการเริ่มต้นการค้นหาใหม่



รูปที่ 3.22 แผนภาพขั้นตอนการทำงานของรีรันหรือรีโพซิชัน

การประยุกต์รีรันหรือรีโพซิชันกับ PSO มีขั้นตอนการทำงานดังรูป 3.22 โดยจะเริ่มจากการสร้างอนุภาคขึ้นมาโดยสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัวภายในปริภูมิของคำตอบ สำหรับจำนวนอนุภาคที่สร้างขึ้นนี้จะไม่มีการเพิ่มหรือลดจำนวนตลอดการทำงาน จากนั้นเป็นขั้นตอนหาค่าความเหมาะสม โดยจะนำอนุภาคทั้งหมดไปหาค่าความเหมาะสมจากตำแหน่งปัจจุบันของอนุภาค สำหรับค่าความเหมาะสมที่ได้นี้จะถูกนำไปใช้ในการปรับปรุง PBEST และ GBEST ซึ่งค่านี้จะใช้ในการคำนวณค่าความเร็วและทิศทางใหม่ของแต่ละอนุภาคเพื่อใช้เคลื่อนที่อนุภาคเหล่านั้นไปยังตำแหน่งใหม่ จากนั้นจะทำการตรวจสอบเงื่อนไขของรีรันหรือรีโพซิชันถ้าตรงตามเงื่อนไขที่กำหนดก็จะดำเนินการรีเซ็ต เช่น เมื่อเกิดปัญหาค่าเหมาะสมที่สุดสัมพันธ์จึงดำเนินการรีเซ็ต เป็นต้น ขั้นตอนต่อมาจะทำการตรวจสอบคำตอบว่าการค้นหาค่าตอบพบหรือไม่ ถ้าพบคำตอบหรือครบจำนวนรอบที่กำหนดเอาไว้ก็จะจบการค้นหา แต่ถ้าไม่ก็กลับไปทำกระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาคในรอบถัดไป

3.2.4 องค์ประกอบของการเริ่มต้นการค้นหาใหม่

สำหรับรีรันและรีโพซิชันทั้งสองวิธีสามารถเรียกรวมว่า รีเซ็ต (Reset) สำหรับองค์ประกอบของรีรันและรีโพซิชัน ทั้งสองวิธีมีองค์ประกอบที่เหมือนกัน ดังนั้นชื่อขององค์ประกอบทั้งสองวิธีจะใช้คำว่ารีเซ็ตแทน โดยมีรายละเอียดขององค์ประกอบทั้งสองวิธีมีดังต่อไปนี้

3.2.4.1 รอบที่จะรีเซ็ต

รอบที่จะรีเซ็ต (Reset Period) คือเงื่อนไขที่ใช้กำหนดว่าเมื่อใดควรดำเนินการรีเซ็ต โดยรอบนี้สามารถกำหนดได้หลายวิธี ตัวอย่างงานวิจัยเช่น

งานวิจัย [50, 52] เสนอกรอบที่จะรีเซ็ตโดยพิจารณาจากรอบของการค้นหาที่มีรัศมีของกลุ่มอนุภาคมากที่สุด (Maximum Swarm Radius หรือ MSR) โดยสรุปมีขั้นตอนดังนี้ การรีเซ็ตจะดำเนินการเมื่อ MSR น้อยกว่าค่าที่กำหนดให้ (Threshold value) โดยสมการการคำนวณค่า MSR คือสมการที่ 3.19

$$r = |x_{\max(\text{fitness})} - x_{\min(\text{fitness})}| \quad (3.19)$$

โดยกำหนดให้

r	คือ รัศมีของกลุ่มอนุภาคมากที่สุด
$x_{\max(\text{fitness})}$	คือ ค่าความเหมาะสมที่ดีที่สุดในกลุ่มประชากร
$x_{\min(\text{fitness})}$	คือ ค่าความเหมาะสมที่แย่ที่สุดในกลุ่มประชากร

งานวิจัย [51] เสนอกรอบที่จะรีเซ็ตโดยพิจารณาจากจำนวนซ้ำของ GBEST ที่ไม่เปลี่ยนแปลงและรัศมีของกลุ่มอนุภาค โดยสรุปมีขั้นตอนดังนี้ ขั้นตอนรีเซ็ตจะดำเนินการเมื่อจำนวนซ้ำของ GBEST มีค่าที่ไม่เปลี่ยนแปลงมากกว่า จำนวนที่กำหนดให้ (MAXSTEP) และรัศมีของกลุ่มอนุภาค (SWARMRADIUS) มีค่าน้อยกว่า รัศมีของขอบที่ถูกกำหนดให้ (BORDERRADIUS) โดยสมการการคำนวณที่ใช้คำนวณรัศมีของกลุ่มอนุภาคและรัศมีของขอบที่ถูกกำหนดให้ คือสมการที่ 3.20 และ 3.21 ตามลำดับ

$$\text{SwarmRadius} = \max_{i=1..M} \left(\sqrt{\sum_{d=1}^D (p_{id} - x_{id})^2} \right) \quad (3.20)$$

โดยกำหนดให้

x_{id}	คือ ตำแหน่งของอนุภาค (i) มิติ (d)
p_{id}	คือ ตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัวของอนุภาค (i) มิติ (d)
D	คือ จำนวนมิติทั้งหมดของแต่ละอนุภาค
M	คือ จำนวนประชากร

$$\text{BorderRadius} = \left(\frac{\sum_{d=1}^D (X_{\max})^2}{1000} \right) \quad (3.21)$$

โดยกำหนดให้

X_{\max}	คือ ความยาวของขอบเขตการค้นหามากที่สุด
D	คือ จำนวนมิติทั้งหมดของแต่ละอนุภาค

งานวิจัย [48] เสนอรอบที่จะรีเซ็ตโดยพิจารณาจากจำนวนซ้ำของค่าความเหมาะสมของอนุภาคเพื่อนบ้านที่ดีที่สุด (Neighborhood Best Position หรือ NBEST) ที่ไม่เปลี่ยนแปลง โดยสรุปมีขั้นตอนดังนี้ ขั้นตอนการรีเซ็ตจะดำเนินการเมื่อจำนวนซ้ำของ NBEST มีค่าไม่เปลี่ยนแปลงมีค่ามากกว่าจำนวนที่กำหนดให้ (MAXSTEP) โดย NBEST นั้นแต่ละอนุภาคจะมีค่าเป็นของตัวเอง นั้นหมายความว่าอนุภาคต่างๆ อาจจะถูกรีเซ็ตไม่พร้อมกัน โดยที่อนุภาคตัวใดตรงตามเงื่อนไขที่กำหนดนี้ก่อน อนุภาคตัวนั้นก็จะถูกดำเนินการรีเซ็ตก่อน

3.2.4.2 สมการรีเซ็ต

สมการรีเซ็ต (Reset Equation) คือสมการที่ประยุกต์ใช้กับการรีเซ็ต โดยวิธีการกระจายอนุภาคนั้น อาจจะใช้วิธีการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบในจำนวนมากๆ [84] หรือ ริโพอิชัน สมการรีเซ็ตจะเหมือนกับกรณีของสมการการกลายพันธุ์ โดยสมการจะถูกนำมาประยุกต์ใช้กับมิติบนตำแหน่งหรือความเร็วของอนุภาค เพื่อที่จะให้เกิดการเปลี่ยนค่าของตำแหน่งหรือความเร็ว ซึ่งจะมีผลกระทบต่อความหลากหลายและการกระจายตัวของกลุ่มอนุภาค

3.2.4.3 จำนวนการรีเซ็ต

จำนวนการรีเซ็ต คือจำนวนอนุภาคที่ถูกดำเนินการรีเซ็ตในแต่ละรอบรีเซ็ต สามารถแบ่งได้สองประเภทคือรีเซ็ตอนุภาคใหม่ทั้งหมด [50] กับรีเซ็ตอนุภาคใหม่เพียงบางส่วน [48, 51]

งานวิจัย [50] เสนอรีเซ็ตอนุภาคใหม่ทั้งหมด คือเมื่อถึงรอบรีเซ็ต ตำแหน่งของอนุภาคทั้งหมดในประชากรจะถูกกำหนดค่าตำแหน่งเริ่มต้นใหม่ทั้งหมด

งานวิจัย [51] เสนอรีเซ็ตอนุภาคใหม่เพียงบางส่วน โดยจำนวนอนุภาคที่จะถูกรีเซ็ตขึ้นอยู่กับค่าความน่าจะเป็นในการรีเซ็ต โดยถ้าค่าที่สุ่มระหว่าง 0 กับ 1 มีค่าน้อยกว่าค่าความน่าจะเป็นนี้ ก็จะดำเนินการรีเซ็ต ซึ่งจำนวนอนุภาคที่จะถูกรีเซ็ตจะมากหรือน้อยขึ้นอยู่กับผู้ออกแบบ ขั้นตอนวิธีที่จะกำหนดค่าความน่าจะเป็นนี้ไว้เท่าใด โดยส่วนมากมักจะมีค่ามาก เช่น 30%, 50%, 70%, 90% ของจำนวนมิติทั้งหมดของทุกอนุภาค เป็นต้น

งานวิจัย [48] เสนอรีเซ็ตอนุภาคใหม่เพียงบางส่วน โดยจำนวนอนุภาคที่จะถูกรีเซ็ตขึ้นอยู่กับรอบรีเซ็ตของอนุภาคตัวเอง ในกรณีนี้คืออนุภาคแต่ละตัวจะมีรอบรีเซ็ตของตัวเอง ถ้าอนุภาคตัวใดถึงรอบที่กำหนดก็จะดำเนินการรีเซ็ต ในขณะที่อนุภาคตัวอื่นอาจจะไม่ดำเนินการรีเซ็ต

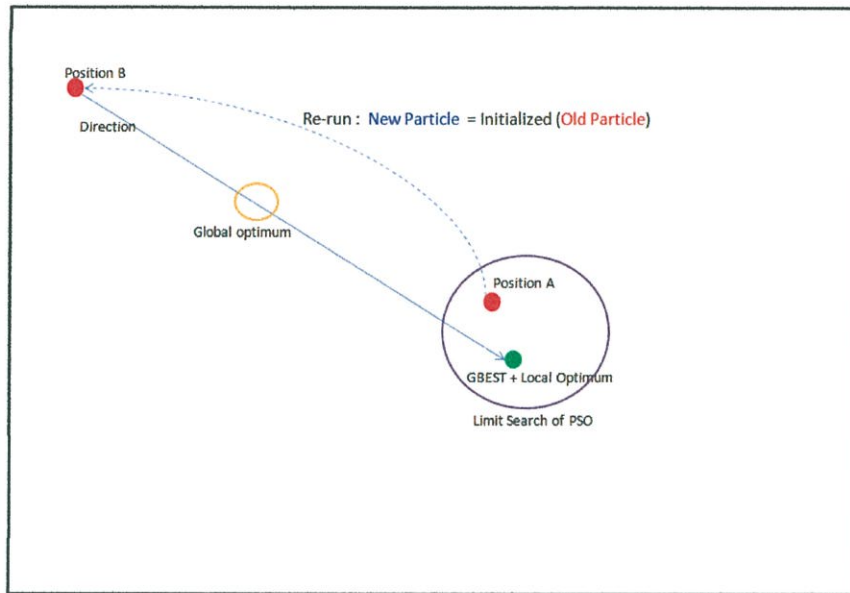
3.2.4.4 ส่วนประกอบที่ถูกรีเซ็ต

ส่วนประกอบที่ถูกรีเซ็ต (Reset Component) คือส่วนประกอบของ PSO เช่น GBEST PBEST ตำแหน่งและความเร็วของอนุภาค ซึ่งส่วนประกอบเหล่านี้จะถูกรีเซ็ต โดยส่วนประกอบที่ถูกรีเซ็ตเหล่านี้จะมีผลกระทบต่อการค้นหาที่แตกต่างกัน และแน่นอนสำหรับการรีเซ็ต ต้องดำเนินการกับตำแหน่งของอนุภาค แต่ความเร็ว GBEST และ PBEST นั้น อาจจะไม่จำเป็นต้องดำเนินการรีเซ็ตเสมอไป โดยการรีเซ็ตค่าเหล่านี้ขึ้นอยู่กับผู้ออกแบบขั้นตอนวิธี ตัวอย่างงานวิจัยเช่น กำหนดให้รีเซ็ตตำแหน่งและความเร็วของอนุภาค [48, 51] และกำหนดให้รีเซ็ตตำแหน่งของอนุภาค เพียงอย่างเดียว [50] เป็นต้น

3.2.5 ข้อดีของการเริ่มต้นการค้นหาใหม่

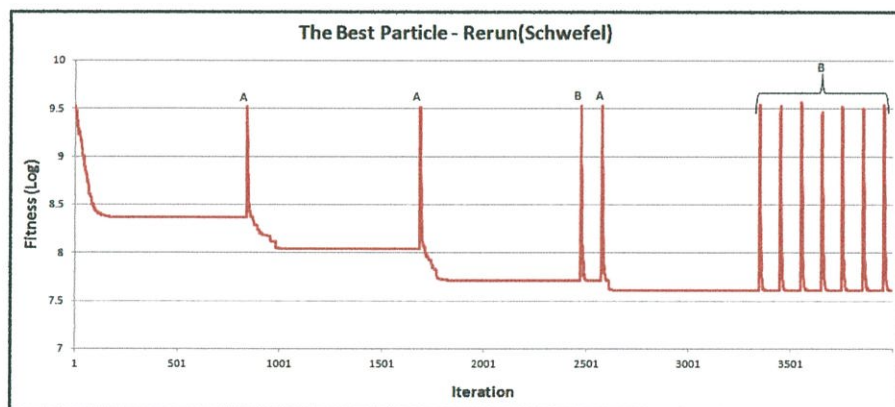
หลักการการทำงานของ PSO อนุภาคทั้งหมดจะเคลื่อนที่เข้าหา GBEST โดยระหว่างทางถ้าพบตำแหน่งที่ดีกว่า GBEST จะเอาตำแหน่งนั้นแทนที่ GBEST และเมื่ออนุภาคเคลื่อนที่เข้าใกล้ GBEST อนุภาคจะค้นหาบริเวณรอบ GBEST เท่านั้น และไม่ออกไปค้นหาในบริเวณอื่นอีก ถ้า GBEST อยู่ที่จุดสุดสัมพัทธ์ มันจะส่งผลให้อนุภาคไม่สามารถพบคำตอบที่ดีกว่าจุดสุดสัมพัทธ์ เสมือนกับการค้นหานั้นสิ้นสุด การค้นหาต่อไปจะสูญเสียเวลาโดยเปล่าประโยชน์ เพื่อไม่ให้สูญเสียเวลาในการค้นหา โดยเปล่าประโยชน์จึงทำการรีเซ็ต โดยการรีเซ็ตจะเพิ่มโอกาสให้ PSO สามารถปรับปรุงคำตอบ อาจจะถูกกล่าวได้ว่า เป็นการเพิ่มโอกาสใช้เวลาในการค้นหาให้คุ้มค่ามากกว่าที่จะปล่อยให้ค้นหาในบริเวณจุดสุดสัมพัทธ์ซึ่งเป็นการสูญเสียเวลาโดยเปล่าประโยชน์ ดังนั้นวิธีการรีเซ็ตจะมีโอกาสในการลองค้นหามากกว่า PSO ที่ไม่ดำเนินการรีเซ็ต จึงได้คำตอบที่ดีกว่า PSO โดยการรีเซ็ตจะเน้นการแก้ปัญหาการติดในจุดสุดสัมพัทธ์ คือเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้นจึงดำเนินการรีเซ็ต

วิธีการรีรันสามารถปรับปรุงคำตอบได้ โดยเมื่อตรวจพบว่าเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น หลังจากนั้นจะดำเนินการกระจายอนุภาคออกจากจุดสุดสัมพัทธ์โดยวิธีการรีรัน การกระจายอนุภาคออกจากกันสามารถเพิ่มความหลากหลายให้กับประชากรเช่นเดียวกับการกลายพันธุ์ แล้วหลังจากนั้นอนุภาคต่างๆ จะเคลื่อนที่กลับเข้าหา GBEST ซึ่งอยู่ในจุดสุดสัมพัทธ์ ถ้าหากระหว่างทางที่อนุภาคเคลื่อนที่กลับนั้นพบตำแหน่งที่ดีกว่า GBEST เดิมก็จะทำการปรับปรุง GBEST เป็นตำแหน่งใหม่ที่พบนั้น ซึ่งขั้นตอนนี้สามารถช่วยปรับปรุงคำตอบให้กับ PSO ได้ แทนที่จะปล่อยให้ PSO นั้นค้นหาคำตอบในจุดสุดสัมพัทธ์ ซึ่งจะทำให้สูญเสียเวลาโดยเปล่าประโยชน์ จะสังเกตได้ว่าวิธีการรีรันสามารถปรับปรุงคำตอบได้โดยใช้หลักการเดียวกับการปรับปรุงคำตอบของ NCM และจากตารางผลการทดลอง 5.12 ซึ่งมาจากผลการทดลองในหัวข้อ 5.5 พบว่า รีรันสามารถเพิ่มประสิทธิภาพของ PSO ให้ได้รับคำตอบที่ดีขึ้น



รูปที่ 3.23 ตัวอย่าง RERUN ช่วยปรับปรุงคำตอบของ PSO

จากรูป 3.23 GBEST (วงกลมสีเขียว) อยู่ที่ตำแหน่งของจุดสุดสัมพัทธ์ (ภายในวงกลมสีม่วง) ทำให้อนุภาคต่างๆ เคลื่อนที่เข้ามารวมกันที่ตำแหน่งจุดสุดสัมพัทธ์และค้นหาบริเวณรอบจุดสุดสัมพัทธ์ ทำให้ไม่สามารถพบจุดสุดสัมบูรณ์ เมื่อเกิดการรีรัน โดยรีรันจะทำการกำหนดค่าเริ่มต้นให้กับอนุภาค (วงกลมสีแดง) อนุภาคกระโดดจากตำแหน่ง A ไปยังตำแหน่ง B หลังจากนั้นอนุภาคจะเคลื่อนที่กลับเข้ามายังจุดสุดสัมพัทธ์ระหว่างทางได้เคลื่อนที่พบกับจุดสุดสัมบูรณ์

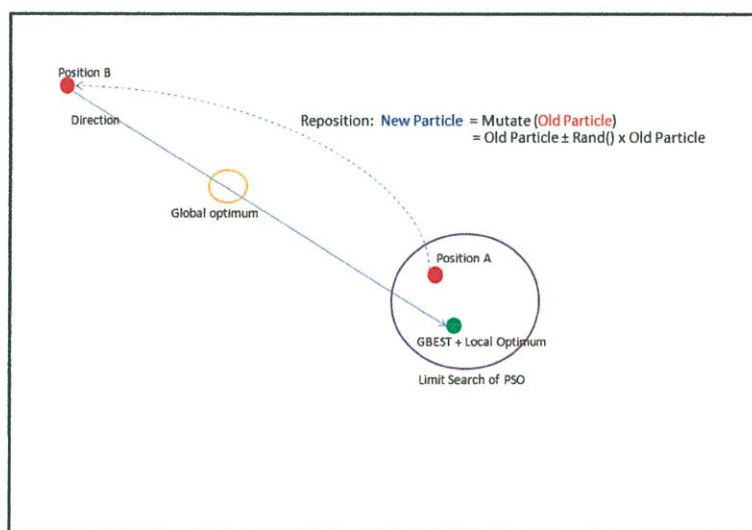


รูปที่ 3.24 ตัวอย่าง RERUN

จากรูป 3.24 ซึ่งมาจากการทดลอง 5.5 เมื่อผ่านการรีเซ็ตเส้นกราฟจะมีค่าเพิ่มขึ้น หลังจากนั้นอนุภาคจะทำการลู่ออกใหม่ เส้นกราฟจึงมีค่าลดลงซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิม ทำให้คำตอบนั้นดีขึ้น ในกรณีที่การรีเซ็ตเริ่มต้นที่จุด A สังเกตได้จากกราฟที่จะลู่ลงไ้มากกว่าก่อนการรีเซ็ต นั้นหมายความว่ากรรีเซ็ตสามารถช่วยปรับปรุงคำตอบของ PSO โดยการกระจายอนุภาคออกจากจุดสุดสัมพัทธ์ ทำให้อนุภาคมีโอกาสได้ค้นหาใหม่อีกครั้ง

3.2.6 ข้อดีของการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม

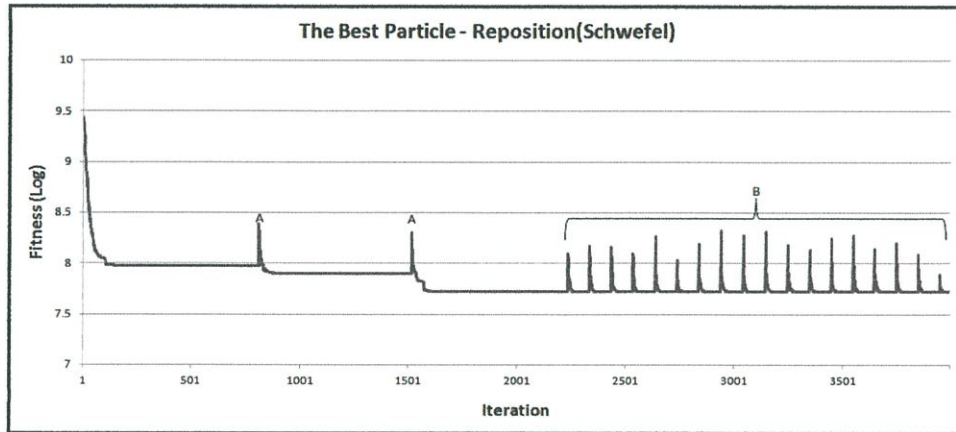
วิธีรีโพอิชชันสามารถปรับปรุงคำตอบของ PSO ด้วยเหตุผลเดียวกับรีรันและ NCM เพราะรีโพอิชชันใช้หลัก NCM เพื่อให้เกิดความหลากหลาย แต่รีโพอิชชันแตกต่าง NCM คือรีโพอิชชันเน้นแก้ปัญหาเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ และเน้นการกระจายอนุภาคออกเป็นบริเวณกว้างโดยขอบเขตการกระจายจะมากหรือน้อยนั้นขึ้นอยู่กับความน่าจะเป็นในการรีโพอิชชันถ้ากำหนดค่านี้มากขอบเขตการกระจายตัวก็จะมาก ถ้ากำหนดค่านี้น้อยขอบเขตการกระจายตัวก็จะน้อย ซึ่งแตกต่างจากรีรันที่ไม่สามารถกำหนดขอบเขตการกระจายตัว และจากตารางผลการทดลอง 5.12 ซึ่งมาจากผลการทดลองในหัวข้อ 5.5 พบว่ารีโพอิชชันสามารถเพิ่มประสิทธิภาพของ PSO ให้รับคำตอบที่ดีขึ้น



รูปที่ 3.25 ตัวอย่าง Reposition ช่วยปรับปรุงคำตอบของ PSO

จากรูป 3.25 GBEST (วงกลมสีเขียว) อยู่ที่ตำแหน่งของจุดสุดสัมพัทธ์ (ภายในวงกลมสีม่วง) ทำให้อนุภาคต่างๆ เคลื่อนที่เข้ามารวมกันที่ตำแหน่งจุดสุดสัมพัทธ์และค้นหาคำตอบบริเวณรอบจุดสุดสัมพัทธ์ ทำให้ไม่สามารถพบจุดสุดสัมบูรณ์ เมื่อเกิดรีโพอิชชันจะทำกรกลายพันธุ์ประเภท NCM ให้กับอนุภาค (วงกลมสีแดง) กระโดดจากตำแหน่ง A ไปยังตำแหน่ง B หลังจากนั้นอนุภาคจะเคลื่อนที่กลับเข้ามายังจุดสุดสัมพัทธ์ระหว่างทางได้เคลื่อนที่พบกับจุดสุดสัมบูรณ์

ผลจากรีรันจะแตกต่างจากรีโพอิชชัน เพราะการกลายพันธุ์หรือรีโพอิชชันจะทำให้มิติบนอนุภาคใดอนุภาคหนึ่งถูกการเปลี่ยนแปลงเพียงบางมิติ ในขณะที่การกำหนดค่าเริ่มต้นใหม่ให้กับอนุภาคหรือรีรัน มิติทั้งหมดบนอนุภาคทั้งหมดจะถูกเปลี่ยนแปลง นอกจากนี้การกลายพันธุ์จะเปลี่ยนค่าบนมิติเพียงเล็กน้อย โดยมิติที่ถูกดำเนินการกลายพันธุ์จะถูกบวกลบในกรณีปัญหาในรูปแบบจำนวนจริงหรือจำนวนเต็ม ถูกกลับบิตในกรณีปัญหาในรูปแบบไบนารี ถูกสลับตำแหน่งในกรณีปัญหาในรูปแบบการสลับลำดับกันในพีชคณิต ซึ่งแตกต่างจากรีรันที่มีมิติที่ถูกดำเนินการสุ่มตำแหน่งใหม่ ซึ่งจะทำให้ค่าบนมิติเปลี่ยนแปลงอย่างมาก ดังนั้นผลลัพธ์จากรีโพอิชชัน จะมีส่วนผสมระหว่างมิติที่เกิดจากการค้นหาเดิม และมิติที่เกิดจากการกลายพันธุ์ ซึ่งจะมีค่ามิติที่ไม่แตกต่างจากเดิมมากนัก ผลจากการค้นหาก่อนรีเซ็ตจึงมีอิทธิพลต่อการค้นหาหลังรีเซ็ต แต่รีรันมิติที่บนอนุภาคจะถูกสุ่มตำแหน่งใหม่เข้าไปแทนทั้งหมด ผลการค้นหาก่อนรีเซ็ตจึงไม่มีอิทธิพลต่อการค้นหาหลังรีเซ็ต



รูปที่ 3.26 ตัวอย่าง Reposition

จากรูป 3.26 ซึ่งมาจากการทดลอง 5.5 เมื่อผ่านการรีเซ็ตเส้นกราฟจะมีค่าเพิ่มขึ้น หลังจากนั้นอนุภาคจะทำการลู่ออกใหม่ ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิม ทำให้ค่าตอบนั้นดีขึ้น ในกรณีที่การรีเซ็ตเริ่มต้นที่จุด A สังเกตได้จากกราฟที่จะลู่ลงไปมากกว่าก่อนการรีเซ็ต นั่นหมายความว่า การรีเซ็ตสามารถช่วยในการปรับปรุงคำตอบของ PSO ด้วยการกระจายอนุภาคออกจากจุดสุดสัมพัทธ์ ทำให้อนุภาคมีโอกาสได้ค้นหาใหม่อีกครั้ง

PSO เมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น ถ้ากรณีจุดสุดสัมบูรณ์อยู่ใกล้กับจุดสุดสัมพัทธ์ วิธีการกลายพันธุ์จะมีโอกาสพบคำตอบที่ดีกว่าวิธีการรีเซ็ต เพราะการกลายพันธุ์ทำให้อนุภาคกระโดดออกไปค้นหารอบจุดสุดสัมพัทธ์ ซึ่งเป็นการค้นหาในบริเวณแคบอย่างละเอียด ดังรูป 3.9 และ 3.12 แต่ถ้ากรณีจุดสุดสัมบูรณ์อยู่ไกลกับจุดสุดสัมพัทธ์ วิธีการกลายพันธุ์อาจจะไม่มีโอกาสพบคำตอบเลย เนื่องจากขอบเขตที่อนุภาคกระโดดออกไปนั้นไปไม่ถึงตำแหน่งที่จุดสุดสัมบูรณ์อยู่ ในขณะที่วิธีการรีเซ็ตทำให้เกิดการกระจายตัวในบริเวณกว้าง จึงเกิดการค้นหาในบริเวณที่กว้างแต่ไม่ละเอียด ซึ่งอาจจะมีโอกาสที่จะเคลื่อนที่ไปพบจุดสุดสัมบูรณ์ได้ ดังรูป 3.23 และ 3.25 ดังนั้นจึงเป็นการยากที่จะสรุปว่าวิธีไหนดีกว่ากัน เนื่องจากวิธีสองจะดีกว่ากันนั้นก็ขึ้นอยู่กับปัญหาที่นำไปแก้ไขว่าจะมีรูปแบบของความซับซ้อนของปัญหาเหมาะสมกับวิธีไหนมากกว่ากัน

3.2.7 ข้อเสียของวิธีการเริ่มต้นการค้นหาใหม่

รีรันช่วยลดปัญหาการติดในจุดสุดสัมพัทธ์ แต่ผลจากรีรันทำให้เกิดปัญหาใหม่ คือ ปัญหาความยุ่งยากในการปรับพารามิเตอร์ของรีรันให้เหมาะสมกับปัญหาที่แก้ไข โดยถ้าดำเนินการรีรันมากเกินไป รีรันจะขัดขวางกระบวนการลู่ออกใหม่ของ PSO เช่นเดียวกับปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบของ NCM ในทางกลับกันถ้าดำเนินการรีรันน้อยเกินไป รีรันก็จะไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ เทคนิครีรันที่เพิ่มเข้าไปก็จะมีประโยชน์เช่นเดียวกับปัญหาการกลายพันธุ์น้อยเกินไปของ NCM และ CM โดยรายละเอียดของปัญหามีดังนี้

จากงานวิจัย [50] ได้เสนอวิธีการรีรัน ซึ่งผลจากการรีรันจะเพิ่มความหลากหลายให้กับประชากร และขยายขอบเขตการค้นหา ขั้นตอนวิธีนี้สามารถช่วยให้เกิดการกระโดดออกจากจุดสุดสัมพัทธ์ อย่างไรก็ตามวิธีนี้จะทำลายโครงสร้างของประชากร เนื่องจากการรีเซ็ตอนุภาคใหม่

ทั้งหมดจะลดความเร็วในการลู่เข้าหากันของอนุภาค และความแม่นยำในการค้นหาคำตอบ [48, 51] แต่ในความจริงแล้วโครงสร้างของประชากรไม่ได้ถูกทำลายทั้งหมด [52] เพราะ GBEST ไม่ถูกรีเซ็ต ซึ่งส่งผลให้การค้นหาในรอบต่อไป อนุภาคทั้งหมดที่อยู่ห่างจาก GBEST เป็นระยะไกลจะลู่เข้าหา GBEST อย่างรวดเร็วเพราะอิทธิพลจากสมการ 2.2 และ 2.3 ในระหว่างทางที่อนุภาคลู่เข้าหา GBEST ถ้าพบตำแหน่งที่ดีกว่า GBEST เดิมก็จะทำการปรับปรุง GBEST เป็นตำแหน่งใหม่ที่พบนั้น ขั้นตอนนี้สามารถช่วยปรับปรุงคำตอบให้กับ PSO ได้ แทนที่จะปล่อยให้อนุภาคค้นหาในจุดสุดสัมพัทธ์ ซึ่งทำให้สูญเสียเวลาโดยเปล่าประโยชน์ อย่างไรก็ตามวิธีนี้มีความยุ่งยากในการกำหนดค่าของพารามิเตอร์ [52] ให้เหมาะสมกับปัญหาที่นำไปแก้ไข โดยพารามิเตอร์ที่ต้องกำหนดนั้นคือ Threshold value ถ้าหากกำหนดค่านี้ไม่เหมาะสมกับปัญหาที่นำไปแก้ไขผลลัพธ์การค้นหาแย่กว่า PSO ที่ไม่ดำเนินการรีเซ็ต จากงานวิจัย [51] ได้เสนอวิธีการรีเซ็ต ซึ่งผลจากการรีเซ็ตจะเพิ่มความหลากหลายของประชากรโดยปราศจากการลดความเร็วของการลู่เข้าหากันของอนุภาค และความแม่นยำในการค้นหาคำตอบ อย่างไรก็ตามวิธีนี้มีความยุ่งยากในการกำหนดค่าของพารามิเตอร์ [52] เพราะมีพารามิเตอร์เพิ่มจาก PSO ปกติอีก 3 ตัว คือ MAXSTEP, BORDERRADIUS, Mutation Probability โดยทั้ง 3 พารามิเตอร์นี้ ถ้ากำหนดค่าไม่เหมาะสมกับปัญหาที่นำไปแก้ไขผลลัพธ์การค้นหาแย่กว่า PSO ที่ไม่ดำเนินการรีเซ็ต ดังนั้นการนำวิธีที่งานวิจัยฉบับนี้ [52] นำเสนอไปประยุกต์ใช้กับปัญหาทั่วไป จึงเป็นเรื่องยุ่งยาก

จากงานวิจัย [48] ได้เสนอวิธีการรีเซ็ต ซึ่งผลจากการรีเซ็ตจะเพิ่มความหลากหลายให้กับกลุ่มอนุภาคและหลีกเลี่ยงปัญหาการลู่เร็วเกินไป อย่างไรก็ตามวิธีนี้มีพารามิเตอร์เพิ่มจาก PSO อีก 1 ตัว คือ MAXSTEP ซึ่งพารามิเตอร์นี้ถ้ากำหนดค่าไม่เหมาะสมกับปัญหาที่นำไปแก้ไขผลลัพธ์การค้นหาแย่กว่า PSO ที่ไม่ดำเนินการรีเซ็ต ดังนั้นการนำวิธีที่งานวิจัยฉบับนี้ [52] นำเสนอไปประยุกต์ใช้กับปัญหาทั่วไปจึงเป็นเรื่องยุ่งยาก

นอกจากนี้ข้อเสียอีกข้อของรีเซ็ต คือเวลาที่สูญเสียจากการลู่เข้าหา GBEST [84] เนื่องจากอนุภาคที่ถูกรีเซ็ต เกิดจากการสุ่มค่าเริ่มต้นใหม่ให้กับทุกมิติของอนุภาค ทำให้อนุภาคกระจายตัวออกไปเป็นบริเวณกว้าง จึงต้องเสียเวลาลู่หากันใหม่เป็นเวลานาน ซึ่งในความจริงแล้วการกระจายตัวมีจุดประสงค์ให้อนุภาคหลุดออกจากจุดสุดสัมพัทธ์แล้วทำให้เกิดการค้นหาในบริเวณอื่น นอกจากจุดสุดสัมพัทธ์เดิม แต่รีเซ็ตทำให้เกิดการกระจายมากเกินไปและสูญเสียเวลาในการลู่เข้ากันของอนุภาคมากเกินไป ผลจากการที่ต้องสูญเสียเวลาในการลู่เข้าหากันของอนุภาค ทำให้จำนวนรอบในการรีเซ็ตน้อยลง โอกาสการลองค้นหาในบริเวณอื่นจึงลดลง ข้อด้อยตรงนี้ทำให้รีเซ็ตไม่มีประสิทธิภาพเท่าที่ควร

3.2.8 ข้อเสียของวิธีการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม

สำหรับข้อเสียของวิธีรีโพอิชชัน คือปัญหาความยุ่งยากในการปรับพารามิเตอร์ของการรีเซ็ตให้เหมาะสมกับปัญหาที่แก้ไขเช่นเดียวกับรีเซ็ต โดยถ้าดำเนินการรีโพอิชชันมากเกินไป รีโพอิชชันจะขัดขวางกระบวนการลู่เข้าหากันของ PSO เช่นเดียวกับปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบของ NCM และรีเซ็ต ในทางกลับกันถ้าดำเนินการรีเซ็ตน้อยเกินไป รีโพอิชชันก็จะไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ เทคนิครีโพอิชชันที่เพิ่มเข้าไปก็จะมีประโยชน์เช่นเดียวกับปัญหาการกลายพันธุ์น้อยเกินไปของ NCM, CM และรีเซ็ต

จุดประสงค์ของการรีเซ็ต คือต้องการให้อนุภาคกระจายตัวออกจากจุดสุดสัมพัทธ์ และค้นหาคำตอบในบริเวณใหม่ ดังนั้นรีโพอิชชันจึงเหมาะสมที่แก้ไขปัญหาการหาจุดที่ดีที่สุดดีกว่ารีรัน เพราะรีโพอิชชันสามารถกำหนดขอบเขตการกระจายตัวของอนุภาคให้มีค่าเท่าที่จำเป็น โดยจากกราฟรูป 3.26 พบว่าเส้นกราฟหลังรีเซ็ตสูงต่ำกว่าเส้นกราฟตอนเริ่มต้น โดยถ้าหากต้องการให้การกระจายตัวน้อยกว่านี้ ก็สามารถกำหนดความน่าจะเป็นในการรีโพอิชชันให้มีค่าน้อยลง ดังนั้นรีโพอิชชันจึงสามารถกำหนดให้เกิดการกระจายตัวเท่าที่จำเป็น ในขณะที่รีรันไม่สามารถกำหนดขอบเขตการกระจายตัวของอนุภาค เพราะมิติของอนุภาคถูกสุ่มค่าเริ่มต้นให้ใหม่ทั้งหมด โดยจากกราฟรูป 3.24 พบว่าเส้นกราฟหลังรีเซ็ตแทบสูงเกือบเท่ากับเส้นกราฟตอนเริ่มต้น นอกจากนี้กราฟทั้งสองแสดงให้เห็นอีกว่าความสูงของกราฟหลังจากรีรันสูงกว่าความสูงกราฟหลังจากรีเซ็ตของรีโพอิชชัน ผลจากการกระจายตัวเท่าที่จำเป็นทำให้ลดเวลาในการลู่ออกกันของอนุภาค ทำให้เพิ่มโอกาสการลองค้นหาในบริเวณอื่น [84] โดยจากตารางผลการทดลอง 5.11 ซึ่งมาจากการทดลอง 5.5 แสดงให้เห็นว่าจำนวนรอบรีเซ็ตของรีโพอิชชันมีจำนวนรอบมากกว่าจำนวนรอบรีเซ็ตของรีรัน ดังนั้นรีโพอิชชันจึงมีโอกาสรองค้นหามากกว่ารีรันจึงทำให้รีโพอิชชันได้รับผลลัพธ์ที่ได้ดีกว่ารีรัน ซึ่งสังเกตผลลัพธ์ได้จากตารางผลการทดลอง 5.12 ซึ่งมาจากการทดลอง 5.5

โดยปกติรีโพอิชชันและรีรันไม่ควรที่จะนำมาประยุกต์ใช้ขณะที่อนุภาคกำลังลู่ออกกัน เพราะผลจากการรีเซ็ตจะเป็นการรบกวนการลู่ออกกันของอนุภาค ซึ่งจะทำให้การค้นหาของ PSO กลายเป็นการค้นหาแบบสุ่ม ในทางกลับกันการรีเซ็ตควรประยุกต์ใช้เมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยการตรวจสอบการเกิดปัญหาการติดในจุดสุดสัมพัทธ์ สามารถกระทำได้ง่าย คือพิจารณาจาก GBEST ที่ไม่เปลี่ยนแปลงเป็นเวลานาน ซึ่งระยะเวลาที่ไม่เปลี่ยนแปลงนี้ควรจะมีค่ามากๆ เพื่อให้มั่นใจว่าเกิดปัญหาการติดในจุดสุดสัมพัทธ์ เมื่อรีเซ็ตดำเนินการในขณะที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ การรีเซ็ตจะไม่ส่งผลให้เกิดเหตุการณ์การค้นหาแบบสุ่ม ดังนั้นปัญหาเรื่องพารามิเตอร์ของรีโพอิชชันและรีรัน จึงสามารถแก้ไขได้ง่าย [84] โดยกำหนดพารามิเตอร์ให้ดำเนินการรีเซ็ตเมื่อตรวจพบว่าเกิดปัญหาการติดในจุดสุดสัมพัทธ์เท่านั้น ซึ่งอาจจะกำหนดค่านี้นให้มีค่ามากๆ เช่น จำนวน GBEST ไม่เปลี่ยนแปลงเป็นเวลา 100, 200, 300, 500 รอบของการค้นหาคำตอบ เป็นต้น แล้วจึงดำเนินการรีเซ็ต

เมื่อเปรียบเทียบประสิทธิภาพรีโพอิชชันมีประสิทธิภาพการค้นหาดีกว่ารีรันโดยจากการทดลองจากตารางการทดลองที่ 5.12 ซึ่งมาจากการทดลอง 5.5 พบว่า รีโพอิชชันมีจำนวนความสำเร็จในการพบคำตอบและค่าเฉลี่ย GBEST มากกว่ารีรันนอกจากนี้ในบางฟังก์ชันรีรันไม่สามารถหาคำตอบพบ ถ้าหาก PSO ไม่สามารถหาคำตอบได้แม้แต่ครั้งเดียว แต่รีโพอิชชันสามารถหาคำตอบพบ เพราะการรีเซ็ตของรีโพอิชชันอนุภาคมีส่วนผสมที่เกิดจากการกลายพันธุ์ จึงทำให้อนุภาคกระจายตัวและมีโอกาสพบกับบริเวณใหม่ที่การค้นหาของ PSO ไม่สามารถพบบริเวณเหล่านี้ ดังนั้นบางปัญหาที่ PSO ไม่สามารถหาคำตอบพบได้แม้แต่ครั้งเดียว แต่รีโพอิชชันสามารถหาคำตอบของปัญหาเหล่านี้พบได้ ด้วยเหตุที่กล่าวมานี้วิทยานิพนธ์ฉบับนี้จึงได้เสนอให้ใช้วิธีรีโพอิชชันเพิ่มประสิทธิภาพการค้นหาของ PSO

บทที่ 4

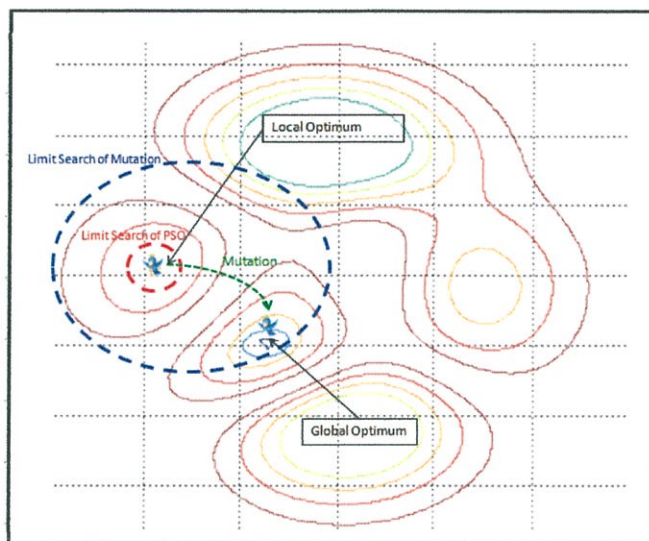
การปรับปรุงการกลายพันธุ์และรีโพอิชัน

การแก้ปัญหาการติดในจุดสุดสัมพัทธ์ที่เกิดขึ้นกับ PSO นักวิจัยส่วนมากได้นำการกลายพันธุ์มาประยุกต์ใช้กับ PSO แต่ผลกระทบจากการกลายพันธุ์ คือปัญหาความยุ่งยากในการปรับพารามิเตอร์ของการกลายพันธุ์ ถ้ากำหนดพารามิเตอร์ให้ดำเนินการกลายพันธุ์มากเกินไป จะทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบสำหรับ NCM และปัญหาการลู่เร็วเกินไปสำหรับ CM ผลกระทบจากปัญหาเหล่านี้ คือการกลายพันธุ์จะทำให้คำตอบจากการค้นหาแย่ลงกว่า PSO ที่ไม่ดำเนินการกลายพันธุ์ ในทางกลับกันถ้ากำหนดพารามิเตอร์ให้การกลายพันธุ์น้อยเกินไป ผลที่ได้จากการค้นหาจะไม่พบคำตอบที่ดีกว่ากระบวนการค้นหาของ PSO

การแก้ปัญหาการติดในจุดสุดสัมพัทธ์ นอกจากการกลายพันธุ์ยังมีอีกวิธีคือรีรัน โดยรีรันมีผลกระทบต่อการค้นหาและผลลัพธ์จากการค้นหาที่แตกต่างจากการกลายพันธุ์คือรีรันสามารถแก้ปัญหาบางปัญหาได้ดีกว่าการกลายพันธุ์ และการกลายพันธุ์ก็สามารถแก้ปัญหาบางปัญหาได้ดีกว่ารีรัน แต่รีรันมีข้อเสียคือเวลาที่สูญเสียจากการลู่เข้าหากันทำให้ผลลัพธ์ที่ได้จากการค้นหาไม่ดีเท่าที่ควร เพื่อลดข้อเสียนี้วิทยานิพนธ์ฉบับนี้จึงเสนอให้ใช้วิธีรีโพอิชันแทนรีรัน เพราะรีโพอิชันสามารถลดระยะเวลาในการลู่เข้าหากันทำให้ลดข้อเสียนี้ แต่หลักการทำงานของรีโพอิชันคือหลังจากกระจายอนุภาคออกแล้ว อนุภาคจะเคลื่อนที่กลับมาที่บริเวณ GBEST เดิมระหว่างทางที่เคลื่อนที่กลับ ถ้าพบคำตอบที่ดีกว่า GBEST เดิมผลจากการรีเซ็ตในรอบนั้นก็ยังสามารถปรับปรุงคำตอบได้ แต่ในความจริงแล้วการรีเซ็ตมีจุดประสงค์ให้อนุภาคเคลื่อนที่เข้าไปค้นหาในบริเวณใหม่ ซึ่งจะได้รับคำตอบที่ดีกว่าการเคลื่อนที่กลับไปค้นหาบริเวณเดิม ดังนั้นรีโพอิชันจึงได้คำตอบไม่ดีเท่าที่ควร เพื่อแก้ข้อเสียนี้วิทยานิพนธ์ฉบับนี้จึงได้เสนอแนวทางปรับปรุงรีโพอิชันเพื่อให้ได้รับคำตอบจากการค้นหาที่ดีขึ้นโดยการรีเซ็ต PBEST และ GBEST เพื่อให้อนุภาคเคลื่อนที่ไปค้นหาคำตอบในบริเวณใหม่

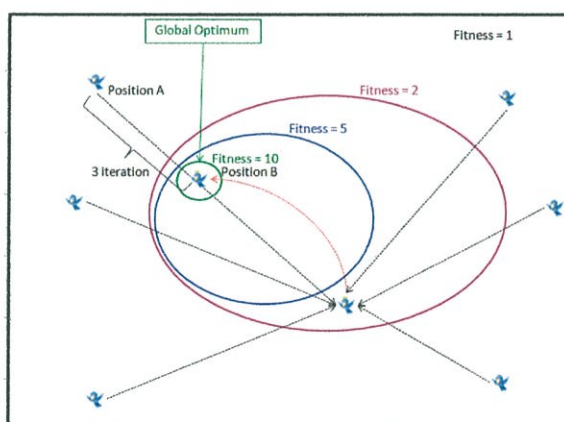
4.1 การเพิ่มประสิทธิภาพของ PSO ที่ประยุกต์ใช้ CM โดยลดปัญหาการลู่เร็วเกินไป

โดยหลักการทำงานของ PSO เมื่ออนุภาคติดอยู่ในจุดสุดสัมพัทธ์จะมีความหลากหลายของประชากรน้อยมาก ดังนั้นเพื่อให้มีความหลากหลายของประชากรเพิ่มขึ้น จึงได้นำวิธีการกลายพันธุ์มาประยุกต์ใช้ร่วมกับ PSO ซึ่งจะเพิ่มขอบเขตการค้นหาของ PSO ให้กว้างขึ้น จึงมีโอกาสพบตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ที่ติดอยู่ดังรูป 4.1 อนุภาคติดอยู่ในจุดสุดสัมพัทธ์ ถ้ากรณีการค้นหาใช้ PSO อย่างเดียว อนุภาคจะค้นหาได้แค่ขอบเขตภายในวงกลมสีแดงเท่านั้น ซึ่งจากรูปพบว่าขอบเขตการค้นหาของ PSO ไม่เพียงพอให้อนุภาคหลุดออกจากจุดสุดสัมพัทธ์ แต่ถ้าการค้นหาใช้ CM อนุภาคจะมีขอบเขตการค้นหาภายในวงกลมสีน้ำเงินซึ่งพบว่า ขอบเขตการค้นหากว้างกว่า PSO มากและอนุภาคมีโอกาสที่จะหลุดออกจากจุดสุดสัมพัทธ์ ดังนั้น CM สามารถเพิ่มประสิทธิภาพการค้นหาของ PSO แล้วได้รับคำตอบที่ดีขึ้น



รูปที่ 4.1 สาคิตขอบเขตของ PSO และขอบเขตของ CM

สำหรับ CM ถ้ากำหนดพารามิเตอร์ให้ดำเนินการกลายพันธุ์น้อยเกินไป จะทำให้เทคนิคการกลายพันธุ์ไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO ดังนั้นการกลายพันธุ์ที่เพิ่มเข้าไปจึงไม่มีประโยชน์ แต่ก็ไม่ส่งผลต่อคำตอบจากการค้นหาจึงไม่เป็นปัญหามากนัก แต่สำหรับ CM ที่กำหนดพารามิเตอร์ให้ดำเนินการกลายพันธุ์มากเกินไป จะทำให้คำตอบจากการค้นหาแยกว่า PSO ปกติที่มีจำนวนอนุภาคมาก เพราะจำนวนของการกลายพันธุ์หนึ่งครั้งเทียบเท่ากับการเคลื่อนที่ของอนุภาคหนึ่งตัวในหนึ่งครั้ง ถ้ากำหนดให้ดำเนินการกลายพันธุ์มากก็ต้องเปรียบเทียบกับ PSO ที่มีอนุภาคจำนวนมาก

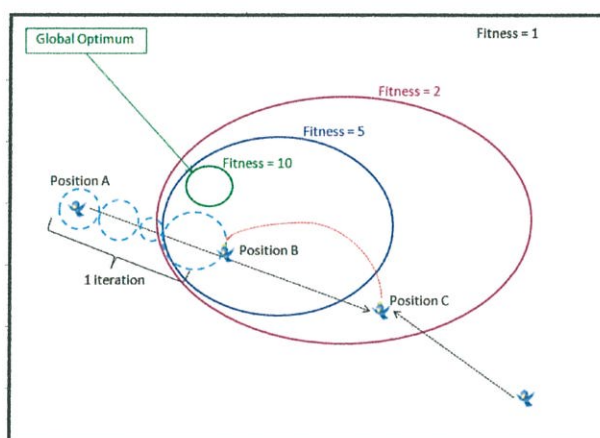


รูปที่ 4.2 สาคิตการลู่เข้าหากันของ PSO

สำหรับ PSO ที่มีจำนวนอนุภาคมากในขณะที่อนุภาคกำลังลู่เข้าหากัน อนุภาคจะเคลื่อนที่จากบริเวณที่กว้างแล้วค่อยแคบลงเรื่อยๆ ดังรูป 4.2 มีจำนวนอนุภาคทั้งหมด 7 อนุภาค สำหรับนกที่สวมมงกุฎเป็นตัวแทนของ GBEST สำหรับปริภูมิของคำตอบมีรายละเอียดดังนี้ คือภายในกรอบสี่ด้านมีความเหมาะสมคือ 1 ภายในวงกลมสีม่วงมีความเหมาะสมคือ 2 ภายในวงกลมสีน้ำเงินมีความเหมาะสมคือ 5 และภายในวงกลมสีเขียวมีความเหมาะสมคือ 10 ซึ่งเป็นจุดสุดสัมบูรณ์ โดยอนุภาคทั้งหมดจะมีทิศทางเคลื่อนที่เข้าหา GBEST เนื่องจากมีจำนวนอนุภาคมากจึงมีโอกาสเคลื่อนที่

ผ่านจุดสุดสัมบูรณ์ โดยอนุภาคที่เริ่มต้นที่ตำแหน่ง A จะเคลื่อนที่ไปหา GBEST ซึ่งอนุภาคนี้ได้พบจุดสุดสัมบูรณ์ที่ตำแหน่ง B โดยใช้เวลา 3 รอบการค้นหาค่าตอบ

สำหรับ CM ที่ได้เปลี่ยนจำนวนอนุภาคเป็นจำนวนที่ดำเนินการกลายพันธุ์ ซึ่งถ้ายังกำหนดให้ดำเนินการกลายพันธุ์มากเท่าใดจำนวนอนุภาคก็จะน้อยลงไปเท่านั้น ถ้าหากกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป ผลจากการกลายพันธุ์จะทำให้อนุภาคแต่ละตัวกลายเป็นการค้นหาแบบละโมบ (Greedy Search) จึงกลายเป็นการค้นหาแบบละโมบหลายจุดแทนการค้นหาแบบ PSO ซึ่งผลจากการค้นหาแบบละโมบนั้นส่งผลเสียต่อการแก้ปัญหาหมัดติโมดอล เพราะการค้นหาแบบละโมบนั้นติดจุดสุดสัมพัทธ์ได้ง่ายมากกว่าการค้นหาแบบ PSO เนื่องจากการค้นหาแบบละโมบไม่มีความสัมพันธ์ระหว่างอนุภาคภายในกลุ่มประชากรและแต่ละอนุภาคไม่ยอมเคลื่อนที่ไปยังตำแหน่งที่แย่กว่าตำแหน่งที่อยู่ปัจจุบัน แต่การค้นหาแบบ PSO อนุภาคสามารถเคลื่อนที่ไปยังตำแหน่งที่แย่กว่าตำแหน่งที่อยู่ปัจจุบัน โดยมีทิศทางเข้าหาตำแหน่งที่ดีที่สุดของกลุ่ม ซึ่งทำให้อนุภาคแต่ละตัวมีโอกาสหลุดออกจากจุดสุดสัมพัทธ์ที่แต่ละตัวนั้นกำลังติดอยู่

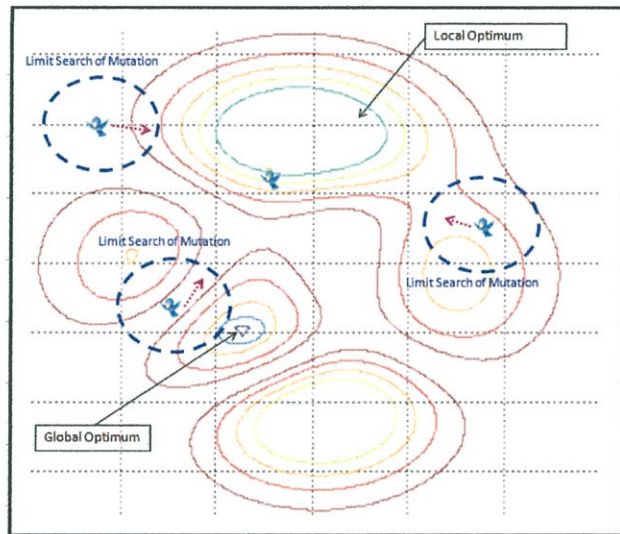


รูปที่ 4.3 สาธิตปัญหาการลู่เร็วเกินไปของ CM

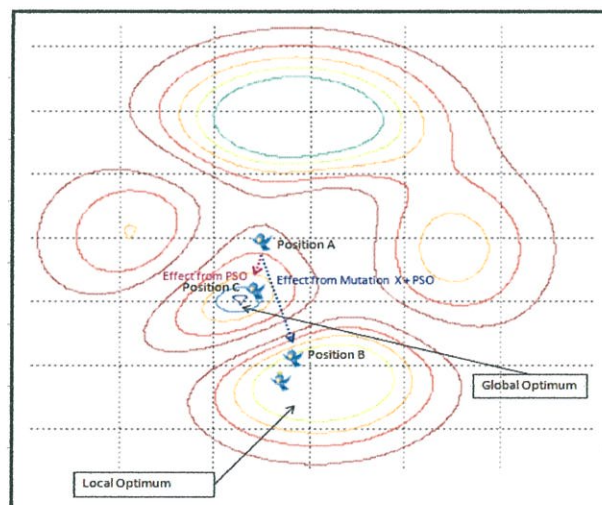
ผลจากการค้นหาแบบละโมบทำให้อนุภาคเคลื่อนที่ไปในทิศทางที่ดีขึ้นเรื่อยๆ จึงมีความเร็วในการลู่เข้าหากันมากกว่าการลู่เข้าหากันของ PSO โดยผลจากการที่อนุภาคลู่เข้าหากันเร็วมากเกินไปนี้ทำให้การค้นหาไม่ละเอียดในระหว่างที่อนุภาคลู่เข้าหากัน ส่งผลให้เกิดปัญหาการลู่เร็วเกินไปที่รุนแรงขึ้นกว่า PSO ที่มีจำนวนการเดาคำตอบเท่ากันดังรูป 4.3 มีจำนวนอนุภาคทั้งหมด 3 อนุภาค ซึ่งสมมติว่าจำนวนการเดาคำตอบในแต่ละรอบการค้นหาค่าตอบเทียบเท่ากับอนุภาค 7 อนุภาคของการค้นหาแบบ PSO สำหรับนกที่สวมมงกุฎเป็นตัวแทนของ GBEST สำหรับปริภูมิของคำตอบมีรายละเอียดเหมือนรูป 4.2 ตอนแรก GBEST อยู่ในตำแหน่ง C และอนุภาคอยู่ในตำแหน่ง A เมื่อดำเนินการกลายพันธุ์ อนุภาคตัวนี้จะทำการเคลื่อนที่แบบละโมบ โดยการค้นหาโดยรอบตัวอนุภาคจะไม่มีรายละเอียด เพราะบางครั้งต้องการกลายพันธุ์หลายครั้งกว่าจะพบตำแหน่งที่ดีกว่าตำแหน่งที่อยู่ปัจจุบัน ซึ่งแทนด้วยวงกลมเส้นประสีฟ้าที่มีขนาดใหญ่ แต่บางครั้งการกลายพันธุ์เพียงไม่กี่ครั้งก็พบตำแหน่งที่ดีกว่าตำแหน่งที่อยู่ปัจจุบัน ซึ่งแทนด้วยวงกลมเส้นประสีฟ้าที่มีขนาดเล็ก เมื่อการกลายพันธุ์พบตำแหน่งที่ดีกว่าตำแหน่งที่อยู่ปัจจุบันก็นำตำแหน่งนั้นแทนที่ตำแหน่งของอนุภาค ทำให้อนุภาคเกิดการเคลื่อนที่ ซึ่งจะวนเช่นนี้ไปเรื่อยๆ จนกว่าจะครบรอบที่ได้กำหนดเอาไว้ จากรูปเพียงหนึ่งรอบการค้นหาค่าตอบ อนุภาคสามารถเคลื่อนที่จากตำแหน่ง A ไปยังตำแหน่ง B แล้วพบกับ

GBEST ใหม่ โดยผลจากการค้นหาแบบละโมภทำให้อนุภาคค้นหาข้ามจุดสุดสัมบูรณ์ แล้วเข้าไปติดจุดสุดสัมพัทธ์อื่นแทน โดยผลจากการที่อนุภาคเข้าสู่หาค้นอย่างรวดเร็วนั้น ส่งผลให้ปัญหาการลู่เร็วเกินไปของ PSO รุนแรงยิ่งขึ้น จึงทำให้ในบางปัญหา CM ได้รับคำตอบจากการค้นหาที่แย่กว่า PSO ที่จำนวนการเดาคำตอบเท่ากัน โดยวิธีการที่กล่าวมานี้คือการประยุกต์ CM กับ GBEST PBEST และตำแหน่งของอนุภาค (X) ซึ่งวิทยานิพนธ์เรียกวิธีนี้ว่า วิธีประยุกต์การกลายพันธุ์ประเภท CM โดยดำเนินการกลายพันธุ์รอบตำแหน่งของอนุภาค แล้วนำผลการกลายพันธุ์มาปรับปรุงตำแหน่งของอนุภาค PBEST และ GBEST หรือ MXUX

โดย MXUX คือวิธีการกลายพันธุ์ที่ถูกนำมาทดลอง เพื่อใช้เป็นตัวเปรียบเทียบกับวิธีการกลายพันธุ์ที่วิทยานิพนธ์ฉบับนี้นำเสนอ นอกจาก MXUX วิทยานิพนธ์ฉบับนี้ได้้นำวิธีการกลายพันธุ์แบบอื่นมาเปรียบเทียบกับวิธีการกลายพันธุ์ที่วิทยานิพนธ์ฉบับนี้นำเสนออีก คือ MGUG และ MPUP สำหรับรายละเอียดของ MGUG และ MPUP จะถูกกล่าวไว้ในภาคผนวก ฉ.1



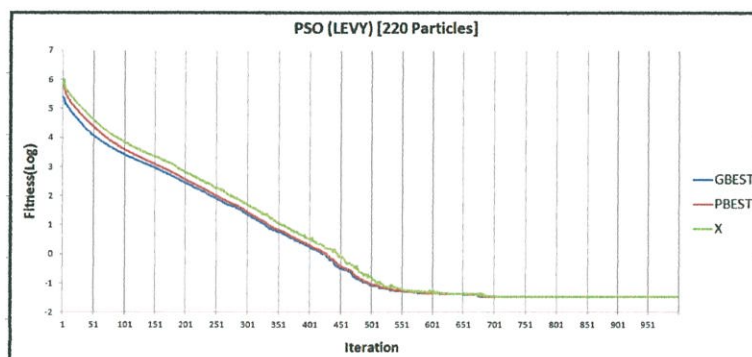
รูปที่ 4.4 วิธีการกลายพันธุ์ของ MXUX



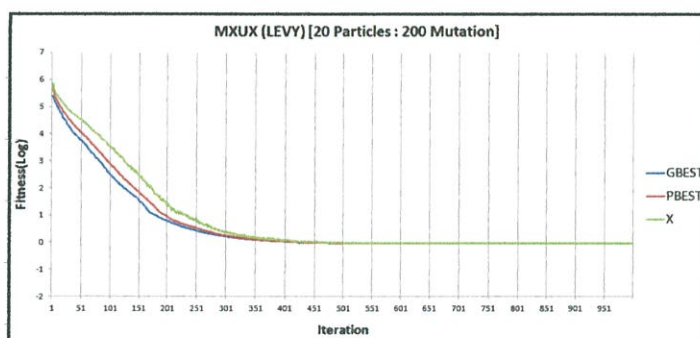
รูปที่ 4.5 วิธีการปรับปรุงตำแหน่งอนุภาคของ MXUX

จากรูป 4.4 MXUX จะดำเนินการรอบอนุภาค โดยมีขอบเขตของการกลายพันธุ์ คือวงกลมเส้นประสีน้ำเงิน ซึ่งการกลายพันธุ์มีการกระจายตัวไปทั่วปริภูมิของคำตอบจึงมีโอกาสพบจุดสุดสัมบูรณ์มากกว่า MGUG เช่นเดียวกับ MPUP

จากรูป 4.5 GBEST อยู่ในตำแหน่งจุดสุดสัมพัทธ์ ผลจากการกลายพันธุ์ CM ทำงานรวมกับการเคลื่อนที่ของ PSO อนุภาคเคลื่อนที่จากตำแหน่ง A ไป B อนุภาคจึงอยู่ในตำแหน่งจุดสุดสัมพัทธ์ร่วมกับ GBEST แล้วติดจุดสุดสัมพัทธ์ แต่การเคลื่อนที่ของอนุภาคโดย PSO ปกติอนุภาคจะค่อยๆ เคลื่อนที่เข้าหา GBEST โดยเปลี่ยนตำแหน่งจาก A ไป C อนุภาคจึงมีโอกาสพบจุดสุดสัมบูรณ์

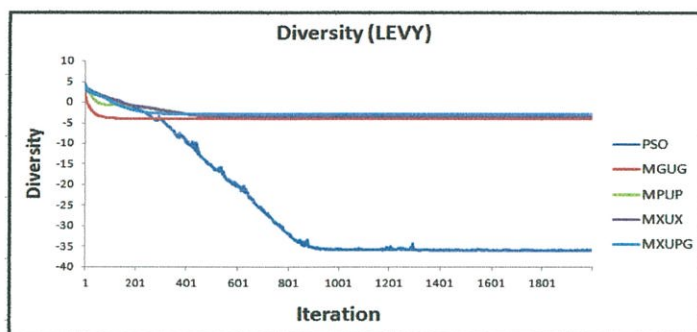


รูปที่ 4.6 การลู่เข้าหากันของ PSO



รูปที่ 4.7 การลู่เข้าหากันของ MXUX

จากกราฟ 4.6 และกราฟ 4.7 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แสดงให้เห็นว่าเส้นกราฟทั้งสามเส้น (PBEST, GBEST, X) ของ MXUX จะเข้าสู่สภาพคงที่เร็วกว่าเส้นกราฟทั้งสามเส้น (PBEST, GBEST, X) ของ PSO นั้นหมายความว่า MXUX มีความเร็วการลู่เข้าหากันเร็วกว่า PSO เช่นเดียวกับ MPUP และ MGUG เพราะแต่ละรอบการค้นหา การเคลื่อนที่ของอนุภาคเกิดจากการขั้นตอนการปรับปรุงตำแหน่งของ PSO และ CM อนุภาคจึงเคลื่อนที่เข้าหากันเร็วกว่า PSO ดังรูป 4.7 จึงเกิดปัญหาการลู่เร็วเกินไปมากกว่า PSO จากเหตุผลเหล่านี้ทำให้บางปัญหา MXUX ได้ผลลัพธ์แย่กว่า PSO ปกติ โดยสังเกตได้จากตารางผลการทดลองที่ 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2



รูปที่ 4.8 ความหลากหลายของประชากรของ PSO, MGUG, MPUP, MXUX และ MXUPG

การกลายพันธุ์ประเภท CM ทั้งสามวิธีสามารถเพิ่มความหลากหลายของประชากร โดยจาก กราฟ 4.8 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.1 แสดงให้เห็นว่าเมื่อเส้นกราฟของ CM ทั้ง 3 วิธี และ PSO เข้าสู่สถานะคงที่ ค่าของ CM ทั้ง 3 วิธีมีค่าใกล้เคียงกันและมากกว่า PSO หมายความว่า เมื่ออนุภาคติดจุดสุดสัมพัทธ์ PSO มีความหลากหลายของประชากรน้อยกว่า CM ทั้ง 3 วิธี ดังนั้น เมื่ออนุภาคติดจุดสุดสัมพัทธ์ PSO มีขอบเขตการค้นห่าน้อยกว่า CM ทั้ง 3 วิธี จึงทำให้ CM ทั้ง 3 วิธี มีโอกาสหลุดออกจากจุดสุดสัมพัทธ์ได้มากกว่า PSO

สำหรับแนวทางการแก้ปัญหาการลู่เร็วเกินไปที่เกิดกับ CM คือพยายามลดความเร็วในการลู่ เข้าหากันของอนุภาคให้ช้าลงหรือใกล้เคียงกับ PSO โดยวิธีการแก้ปัญหานี้สามารถทำได้สองวิธี

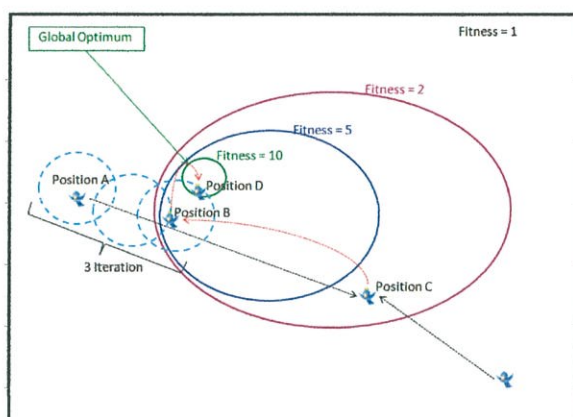
วิธีแรกกำหนดให้ดำเนินการกลายพันธุ์น้อยลง แล้วเพิ่มจำนวนอนุภาคทดแทนส่วนที่หายไป โดยกำหนดสัดส่วนให้เหมาะสมกับปัญหาที่ถูกแก้ไข ก็จะสามารถแก้ปัญหาการลู่เร็วเกินไปที่เกิดจาก CM ได้ ซึ่งการแก้ปัญหาโดยการปรับพารามิเตอร์ให้เหมาะสมกับปัญหาที่ถูกแก้ไขนั้นเป็นเรื่องที่ ยุ่งยาก เพราะแต่ละปัญหาต้องกำหนดพารามิเตอร์ไม่เท่ากันวิธีนี้จึงไม่ค่อยเหมาะสม

วิธีที่สองคือการกลายพันธุ์จะไม่ทำให้เกิดการเปลี่ยนแปลงตำแหน่งของอนุภาคจะเกิดจาก ขั้นตอนการทำงานของ PSO เท่านั้น สำหรับการกลายพันธุ์จะเป็นตัวช่วยเพิ่มความหลากหลายให้กับ ประชากร โดยไม่รบกวนการทำงานของ PSO ซึ่งจะทำให้ความเร็วของการลู่เข้าหากันใกล้เคียงกับ PSO ทำให้แทบจะไม่เพิ่มผลกระทบจากปัญหาการลู่เร็วเกินไป

จากหลักการที่ได้กล่าวมานี้เป็นที่มาของการออกแบบขั้นตอนวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ โดยให้ PSO เป็นตัวดำเนินการเปลี่ยนแปลงตำแหน่งของอนุภาคเท่านั้น สำหรับผลการกลายพันธุ์ต้อง ไม่นำมาประยุกต์ใช้เปลี่ยนแปลงตำแหน่งของอนุภาคโดยตรง แต่ผลเหล่านี้จะใช้ปรับปรุงส่วนที่ใช้ บันทึกราคาตำแหน่งที่ดีที่สุดจากการค้นหา คือ PBEST และ GBEST แทน และให้ PSO เป็นตัวดำเนินการ เคลื่อนที่อนุภาคให้เข้าหา PBEST และ GBEST ตามหลักการทำงานของ PSO ดังรูป 4.9 มีจำนวน อนุภาคทั้งหมด 3 อนุภาค ซึ่งสมมติว่าจำนวนการเดาคำตอบในแต่ละรอบการค้นหาคำตอบเท่ากับ อนุภาค 7 อนุภาคของการค้นหาแบบ PSO ปกติ สำหรับนกที่สวมมงกุฎเป็นตัวแทนของ GBEST สำหรับปริภูมิของคำตอบมีรายละเอียดเหมือนกันรูป 4.2 ตอนแรก GBEST อยู่ในตำแหน่ง C และ อนุภาคอยู่ในตำแหน่ง A เมื่ออนุภาคถูกดำเนินการกลายพันธุ์ อนุภาคตัวนี้จะไม่ได้เคลื่อนที่แบบ ละโมบ เพราะการกลายพันธุ์ไม่ได้ปรับปรุงตำแหน่งของอนุภาคโดยตรง แต่ผลจากการกลายพันธุ์จะใช้ ปรับปรุง PBEST ของอนุภาคนั้น และ GBEST จึงทำให้การค้นหารอบอนุภาคตัวนี้มีบริเวณกว้างและมี ความละเอียดมากในระหว่างที่อนุภาคกำลังลู่เข้าหากัน โดยแทนด้วยวงกลมเส้นประสีฟ้าที่มีขนาด ใหญ่ หลังจากดำเนินการกลายพันธุ์เสร็จอนุภาคจะเคลื่อนที่ตามหลักการทำงานของ PSO โดย

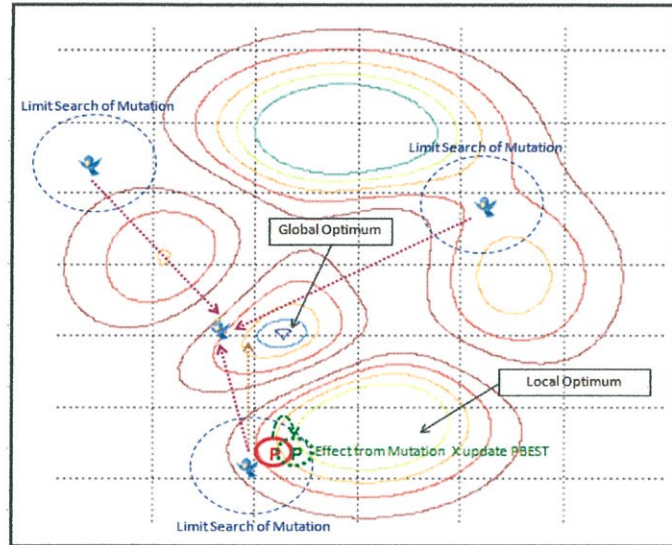
เคลื่อนที่จากตำแหน่ง A ไปตำแหน่ง B และใช้เวลา 3 รอบการค้นหาคำตอบซึ่งเท่ากับ PSO ปกติ เมื่ออนุภาคอยู่ในตำแหน่ง B อนุภาคมีขอบเขตการค้นหาที่กว้างจึงมีโอกาสพบกับจุดสุดสัมบูรณ์

ผลจากที่อนุภาคไม่ได้เคลื่อนที่แบบละโมบ ทำให้ขอบเขตการค้นหาของอนุภาคแต่ละตัวมีบริเวณกว้างขึ้น และทำให้การลู่เข้าหากันของอนุภาคช้าลงหรือเทียบเท่ากับ PSO เพราะผลจากการกลายพันธุ์หรือการค้นหาแบบละโมบไม่ชักนำให้อนุภาคเคลื่อนที่เข้าหากันเร็วกว่า PSO จึงไม่ทำให้เกิดปัญหาการลู่เร็วเกินไปรุนแรงขึ้นเมื่อเปรียบเทียบกับ PSO ดังนั้นคำตอบที่ได้รับจากการค้นหาของขั้นตอนวิธีที่นำเสนอจึงไม่แย่กว่า PSO

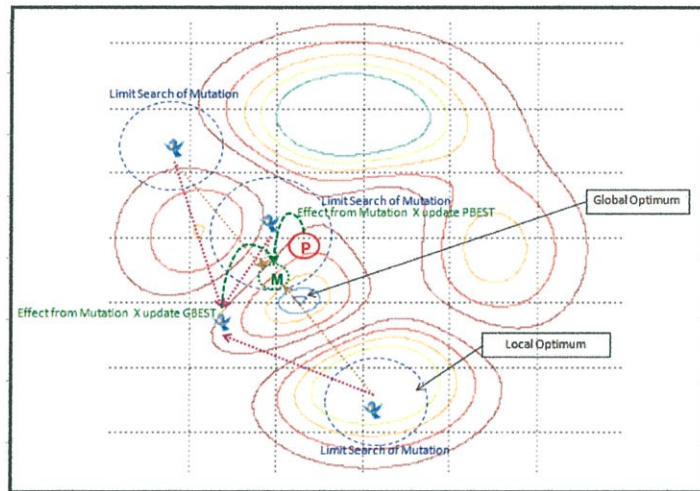


รูปที่ 4.9 สาธิตแนวทางการแก้ปัญหาการลู่เร็วเกินไปของ CM

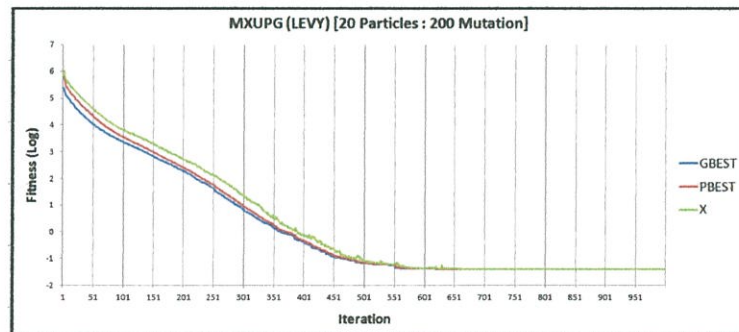
จากแนวคิดที่ต้องการเพิ่มความหลากหลายของประชากรให้กับ PSO แต่ให้ความเร็วในการลู่เข้าหากันของอนุภาคใกล้เคียงกับ PSO ดังนั้นวิธยานิพนธ์ฉบับนี้จึงได้ออกแบบขั้นตอนวิธีที่มีขั้นตอนการทำงานดังนี้ ดำเนินการกลายพันธุ์รอบตำแหน่งของอนุภาคในระหว่างทางที่อนุภาคลู่เข้าหา GBEST ถ้าผลการกลายพันธุ์พบค่าที่ดีกว่า PBEST ก็เอาค่าที่พบนั้นแทนค่า PBEST การปรับปรุง PBEST มีผลเปลี่ยนแปลงเส้นทางการเคลื่อนที่ของอนุภาคที่ถูกการกลายพันธุ์ดังรูป 4.10 ถ้าผลการกลายพันธุ์พบค่าที่ดีกว่า GBEST ก็เอาค่าที่พบนั้นแทนค่า GBEST การปรับปรุง GBEST มีผลเปลี่ยนแปลงเส้นทางการเคลื่อนที่ของอนุภาคทั้งหมดในรอบการค้นหาต่อไปดังรูป 4.11 วิธีนี้จะไม่เหมือนกับวิธี MXUX ที่นำผลการกลายพันธุ์มาปรับปรุงตำแหน่งของอนุภาคโดยตรง แต่นำผลการกลายพันธุ์มาปรับปรุง PBEST และ GBEST ซึ่งทำให้เกิดการเปลี่ยนแปลงตำแหน่งของอนุภาคทางอ้อมคืออนุภาคมีทิศทางเบนเข้าหาจุดที่ถูกเปลี่ยนแปลงจึงเกิดการเปลี่ยนแปลงเส้นทางการค้นหา หลังจากดำเนินการกลายพันธุ์เรียบร้อยแล้วขั้นตอนต่อมาก็จะทำการปรับปรุงตำแหน่งของอนุภาคโดยขั้นตอนการทำงานของ PSO ปกติ สำหรับรายละเอียดและเหตุผลที่วิธีนี้ไม่ทำให้เกิดปัญหาการลู่เร็วเกินไปรุนแรงขึ้นกว่า PSO ปกติ ได้ถูกเขียนไว้ก่อนหน้านี้แล้ว สำหรับวิธีนี้ถูกเรียกว่าการดำเนินการกลายพันธุ์กับตำแหน่งของอนุภาคและผลการกลายพันธุ์ใช้ปรับปรุง PBEST และ GBEST (PSO with mutation X and update PBEST and GBEST หรือ MXUPG) ซึ่งเป็นวิธีที่วิธยานิพนธ์ฉบับนี้นำเสนอ



รูปที่ 4.10 วิธีการกลายพันธุ์ของ MXUPG และการปรับปรุง PBEST



รูปที่ 4.11 วิธีการกลายพันธุ์ของ MXUPG และการปรับปรุง PBEST และ GBEST



รูปที่ 4.12 การลู่เข้าหากันของ MXUPG

จากรูป 4.6 และ รูป 4.12 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แสดงให้เห็นว่าเส้นกราฟทั้งสามเส้น (GBEST, PBEST, X) ของ MXUPG จะเข้าสู่สภาพคงที่ใกล้เคียงเส้นกราฟทั้งสามเส้น (GBEST, PBEST, X) ของ PSO หมายความว่า MXUPG มีความเร็วการลู่เข้าหากันใกล้เคียง PSO ดังนั้น MXUPG จึงไม่เพิ่มปัญหาการลู่เร็วเกินไปเมื่อเปรียบเทียบกับ PSO

จากรูป 4.8 แสดงให้เห็นว่า MXUPG มีความหลากหลายของประชากรใกล้เคียงกับ CM ทั้ง 3 วิธี และมีค่ามากกว่า PSO โดย PSO มีจำนวนในการเดาคำตอบเท่ากับ MXUPG และ CM ทั้ง 3 วิธี ดังนั้นเมื่อ PSO ติดอยู่ในจุดสุดสัมพัทธ์ อนุภาคจะไม่สามารถกระโดดออกจากจุดสุดสัมพัทธ์ เพราะมีขอบเขตการค้นหาที่น้อย แต่ MXUPG และ CM ทั้ง 3 วิธี เมื่อติดอยู่ในจุดสุดสัมพัทธ์ อนุภาคมีโอกาสที่จะกระโดดออกจากจุดสุดสัมพัทธ์มากกว่า PSO เพราะมีขอบเขตการค้นหาที่กว้าง

จากรูป 4.10 การกลายพันธุ์แบบ MXUPG การกลายพันธุ์จะดำเนินการรอบอนุภาค โดยมีขอบเขตของการกลายพันธุ์ คือวงกลมเส้นประสีน้ำเงิน ผลจากการกลายพันธุ์นำมาเปรียบกับ PBEST คือวงกลมสีแดง ตำแหน่งที่ได้จากการกลายพันธุ์ดีกว่า PBEST เดิม PBEST ก็จะเปลี่ยนตัวเองเป็นตำแหน่งที่ได้จากการกลายพันธุ์นั้นคือวงกลมเส้นประสีเขียว ผลจากการเปลี่ยนตำแหน่งของ PBEST ส่งผลให้อนุภาคตัวนั้นเกิดการเปลี่ยนแปลงเส้นทาง โดยเส้นทางเดิมคือเส้นประสีม่วงและเส้นทางใหม่คือเส้นประสีน้ำตาล และการกลายพันธุ์มีการกระจายไปทั่วพื้นที่จึงมีโอกาสพบจุดสุดสัมบูรณ์มากกว่า MGUG เช่นเดียวกับ MPUP และ MXUX

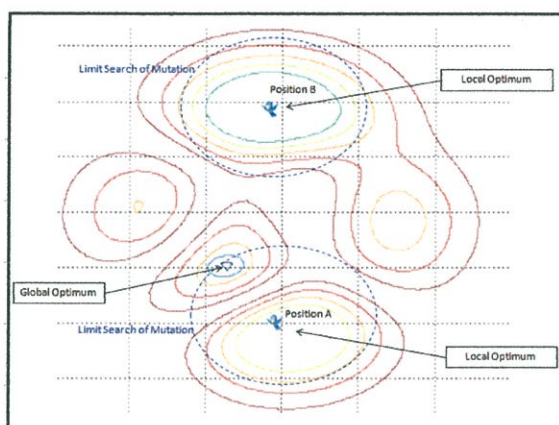
จากรูป 4.11 การกลายพันธุ์แบบ MXUPG การกลายพันธุ์จะดำเนินการรอบอนุภาค โดยมีขอบเขตของการกลายพันธุ์ คือวงกลมเส้นประสีน้ำเงิน PBEST คือวงกลมสีแดง และนกที่สวมมงกุฎคือ GBEST ผลจากการกลายพันธุ์นำมาเปรียบกับ PBEST และ GBEST โดยตำแหน่งที่ได้จากการกลายพันธุ์ดีกว่า PBEST และ GBEST จึงเปลี่ยนตัวเองเป็นตำแหน่งที่ได้จากการกลายพันธุ์คือวงกลมเส้นประสีเขียว ผลจากการเปลี่ยนตำแหน่งของ GBEST ส่งผลให้อนุภาคทั้งหมดเกิดการเปลี่ยนแปลงเส้นทาง โดยเส้นทางเดิมคือเส้นประสีม่วงและเส้นทางใหม่คือเส้นประสีน้ำตาล

จากที่กล่าวมานี้ MXUPG สามารถเพิ่มความหลากหลายของประชากร จึงช่วยแก้หรือลดปัญหาการติดจุดสุดสัมพัทธ์และ MXUPG แทบจะไม่เพิ่มความเร็วในการลู่เข้าหากัน จึงไม่เพิ่มปัญหาการลู่เร็วเกินไป ผลลัพธ์จากการค้นหาจึงดีกว่า PSO

4.2 การเพิ่มของประสิทธิภาพของ PSO หลังจากเกิดปัญหาการติดในจุดสุดสัมพัทธ์

4.2.1 การติดในจุดสุดสัมพัทธ์ของ PSO ที่ประยุกต์ใช้ CM

สำหรับการค้นหาคำตอบด้วย MXUPG เมื่อ GBEST ติดอยู่ในจุดสุดสัมพัทธ์การกลายพันธุ์ช่วยเพิ่มขอบเขตการค้นหาจุดสุดสัมพัทธ์ให้กว้างขึ้นเมื่อเปรียบเทียบกับ PSO ถ้าขอบเขตของการกลายพันธุ์กว้างพอที่จะพบตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ การกลายพันธุ์ก็จะสามารถช่วยให้หลุดออกจากจุดสุดสัมพัทธ์ได้ดังรูป 4.13 ในตำแหน่ง A แต่ถ้าขอบเขตของการกลายพันธุ์ไม่กว้างพอที่จะพบตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ อนุภาคจะไม่สามารถค้นหาตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ที่ติดอยู่และจะติดอยู่ในจุดสุดสัมพัทธ์นั้นตลอดไป ในกรณีนี้การกลายพันธุ์ไม่สามารถแก้ปัญหาการติดจุดสุดสัมพัทธ์ได้ดังรูป 4.13 ในตำแหน่ง B ซึ่งเหตุการณ์นี้เรียกว่าปัญหาขอบเขตการกลายพันธุ์ (Limit Mutation Problem)



รูปที่ 4.13 สาธิตปัญหาขอบเขตการกลายพันธุ์

จากรูป 4.13 ถ้าอนุภาคติดจุดสุดสัมพัทธ์ในตำแหน่ง A ขอบเขตของการกลายพันธุ์ครอบคลุมจุดสุดสัมบูรณ์ เมื่อดำเนินการกลายพันธุ์ไปเรื่อยๆ จะมีโอกาสพบจุดสุดสัมบูรณ์ แต่ถ้าอนุภาคติดจุดสุดสัมพัทธ์ในตำแหน่ง B ไม่ว่าจะดำเนินการกลายพันธุ์มากแค่ไหนก็ตาม การกลายพันธุ์ไม่มีโอกาสที่อนุภาคจะพบจุดสุดสัมบูรณ์ เนื่องจากขอบเขตการกลายพันธุ์ไม่ครอบคลุมจุดสุดสัมบูรณ์ ซึ่งเหตุการณ์นี้เรียกว่าปัญหาขอบเขตการกลายพันธุ์ โดยรายละเอียดเพิ่มเติมจะถูกกล่าวไว้ในภาคผนวก ฅ.2

4.2.2 การปรับปรุงวิธีการเริ่มต้นการค้นหาในบริเวณใหม่ที่กำหนดจากจุดเดิม

ทั้ง MXUPG และ PSO ประสบปัญหาการติดในจุดสุดสัมพัทธ์ โดย MXUPG จะเรียกปัญหาการติดในจุดสุดสัมพัทธ์นี้ว่าปัญหาขอบเขตการกลายพันธุ์ ซึ่งผลกระทบเสมือนกับการค้นหานั้นสิ้นสุดการค้นหาโดยการค้นหาต่อไปจะไม่มีประโยชน์ เพื่อให้การค้นหาต่อไปมีประโยชน์จึงได้นำวิธีการรีโพซิชันเข้ามาประยุกต์ใช้

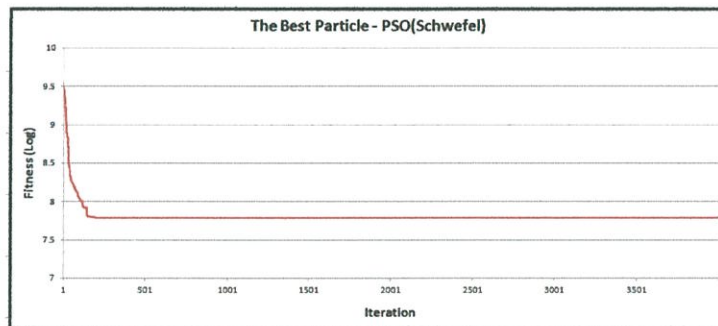
โดยจุดประสงค์ของรีโพซิชันหรือรีรัน คือแก้ปัญหาการติดในจุดสุดสัมพัทธ์โดยให้อนุภาคกระจายตัวออกจากขอบเขตจุดสุดสัมพัทธ์แล้วไปค้นหาบริเวณอื่นนอกจากจุดสุดสัมพัทธ์เดิม โดยการดำเนินการแบบรีรันได้ดำเนินการเกินความจำเป็น เพราะรีรันจะรีเซ็ตตำแหน่งของอนุภาคทั้งหมดส่งผลให้เกิดการกระจายตัวเป็นบริเวณกว้างอนุภาคต้องใช้เวลาอันยาวนานเพื่อลู่เข้าหากันใหม่ จึงสูญเสียวเวลาไปกับการลู่เข้าหากันทำให้จำนวนรอบของการรีเซ็ตลดลงและทำให้โอกาสการค้นหาในบริเวณอื่นลดลง

เพื่อเพิ่มประสิทธิภาพของรีรันจึงจำเป็นต้องทำให้เวลาการลู่เข้าหากันของอนุภาคลดลง วิทยานิพนธ์ฉบับนี้จึงได้เสนอให้ใช้รีโพซิชันแทนรีรัน เพราะรีโพซิชันสามารถกำหนดขอบเขตการกระจายตัวของอนุภาคจึงสามารถทำให้อนุภาคกระจายตัวออกจากกันเท่าที่จำเป็น ส่งผลให้อัตราการลู่เข้าหากันเร็วกว่ารีรัน ส่งผลให้จำนวนรอบของการรีเซ็ตมากกว่ารีรันจึงทำให้โอกาสการค้นหาในบริเวณอื่นมากกว่ารีรัน

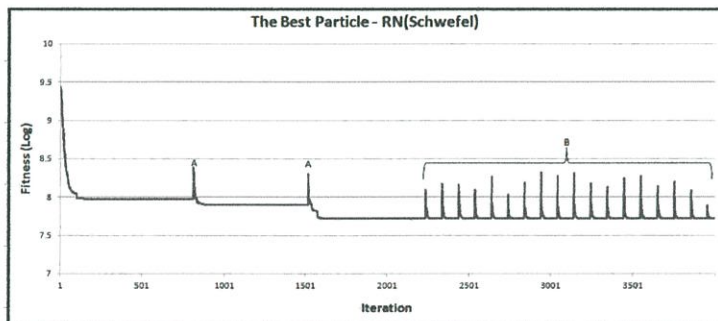
รีโพซิชันเมื่อรีเซ็ตแล้วอนุภาคจะกระจายตัวออกจากกัน หลังจากการรีเซ็ตอนุภาคจะดำเนินการค้นหาด้วย PSO อนุภาคจะเคลื่อนที่กลับเข้ามารวมกันที่ GBEST เดิม เสมือนว่าการค้นหาไม่ได้ค้นหาในบริเวณใหม่แต่กลับเคลื่อนกลับเข้ามาค้นหาในบริเวณเดิม เพียงแต่ระหว่างทางที่

เคลื่อนที่กลับเข้ามาหา GBEST อาจจะมีตำแหน่งที่ดีกว่า GBEST เดิมจึงสามารถปรับปรุงคำตอบ แต่การไม่รีเซ็ต GBEST ทำให้อนุภาคยังคงลู่เข้าหาบริเวณเดิม ซึ่งแทนที่ลู่เข้าไปค้นหาคำตอบใน บริเวณใหม่ๆ โดยจุดประสงค์ของการรีเซ็ตคือต้องการให้อนุภาคไปค้นหาบริเวณใหม่มากกว่าค้นหา ระหว่างทางที่ลู่เข้าหาบริเวณเดิม ดังนั้นการไม่รีเซ็ต GBEST ทำให้คำตอบที่ได้ไม่ดีเท่าที่ควร เมื่อ เปรียบเทียบกับการรีเซ็ต GBEST และสำหรับกรณีไม่รีเซ็ต PBEST ตามหลักการทำงานของ PSO จะ นำ PBEST ที่ดีที่สุดมาแทน GBEST จึงทำให้ได้ผลลัพธ์เช่นเดียวกับการไม่รีเซ็ต GBEST

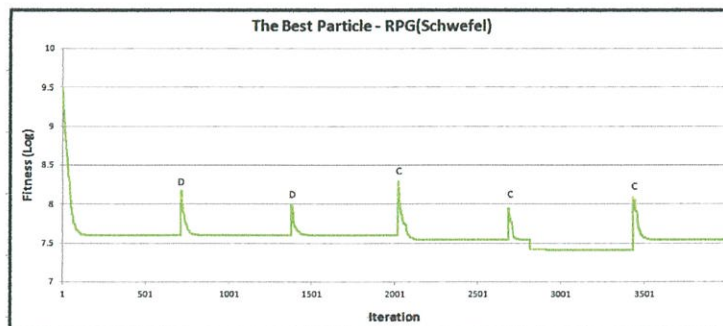
เพื่อแสดงให้เห็นผลกระทบจากการไม่รีเซ็ต PBEST และ GBEST วิทยานิพนธ์ฉบับนี้ จึงได้ทำ การทดลอง โดยแบ่งกรณีทดลองออกเป็นดังนี้ รีโพอิชันที่ไม่รีเซ็ต PBEST และ GBEST (Reposition with non-reset หรือ RN) และรีโพอิชันที่รีเซ็ต PBEST และ GBEST (Reposition with reset PBEST and GBEST หรือ RPG) สำหรับการทดลองที่รีโพอิชันจะถูกรีเซ็ตเพียงบางส่วน (RG หรือ RP) ซึ่งจะถูกนำมาใช้ทดลองเปรียบเทียบกับ RPG นั้น สำหรับรายละเอียดของ RG และ RP จะถูกกล่าวไว้ในภาคผนวก ฅ.3



รูปที่ 4.14 ตัวอย่างของ PSO



รูปที่ 4.15 ตัวอย่างของ RN

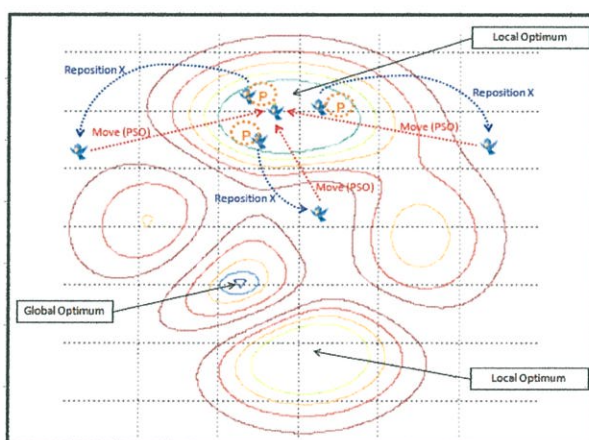


รูปที่ 4.16 ตัวอย่างของ RPG

จากกราฟ 4.14 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 เส้นกราฟในช่วงแรกจะมีค่าลดลงเนื่องจากการค้นหาด้วย PSO ทำให้ค่าความเหมาะสมของอนุภาคดีขึ้น หลังจากนั้นเส้นกราฟจะเข้าสู่สภาพคงที่เนื่องจากเกิดปัญหาการติดในจุดสุดสัมพัทธ์จนสิ้นสุดการค้นหา

จากกราฟ 4.15 และ 4.16 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 เส้นกราฟในช่วงแรกจะมีค่าลดลงเนื่องจากการค้นหาด้วย PSO ทำให้ค่าความเหมาะสมดีขึ้น หลังจากนั้นเส้นกราฟจะเข้าสู่สภาพคงที่จนกระทั่งสามารถสรุปได้ว่าเวลานั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น หลังจากนั้นจะดำเนินการรีเซ็ตซึ่งจะทำให้เส้นกราฟสูงขึ้นเนื่องจากอนุภาคถูกกระจายตัวออกจากกันค่าความเหมาะสมของอนุภาคจึงแย่งลง หลังจากการรีเซ็ตจะค้นหาด้วย PSO ปกติจึงทำให้เส้นกราฟลดลง

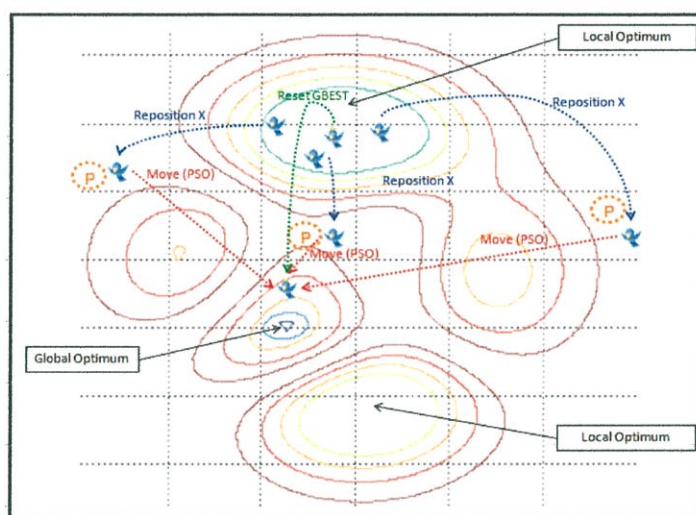
จากกราฟ RN อนุภาคจะเคลื่อนที่จากตำแหน่งที่มันกระจายตัวออกไปแล้วมุ่งเข้ามากันมายังตำแหน่ง PBEST และ GBEST เดิมก่อนการรีเซ็ต ในระหว่างทางที่มันเคลื่อนที่กลับ ถ้าอนุภาคได้พบตำแหน่งที่ดีกว่า GBEST เดิม จะทำให้พบ GBEST ใหม่ จึงสามารถปรับปรุงคำตอบให้ดีขึ้นได้ ซึ่งสามารถสังเกตได้จากรอบรีเซ็ตที่เริ่มต้นที่จุด A จากกราฟ 4.15 และจากตารางผลการทดลองที่ 5.13 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 แสดงให้เห็นว่า RN ได้รับความตอบการค้นหาดีกว่า PSO ดังนั้นทั้ง 3 วิธี สามารถปรับปรุงคำตอบให้ดีขึ้นและเพิ่มประสิทธิภาพการค้นหาของ PSO แต่โดยส่วนมากระหว่างทางที่อนุภาคจะมุ่งเข้ามา GBEST เดิมจะไม่พบตำแหน่งที่ดีกว่า GBEST เดิม จึงไม่สามารถปรับปรุงคำตอบให้ดีขึ้น ซึ่งสามารถสังเกตได้จากรอบรีเซ็ตที่เริ่มต้นที่จุด B จากกราฟ 4.15 ที่เป็นเช่นนี้มาจากสาเหตุที่การค้นหาที่ถูกบังคับให้ทิศทางลู่กลับเข้ามา GBEST เดิม จึงทำให้การค้นหาที่มีขอบเขตที่แคบอยู่ในบริเวณรอบ GBEST เดิมซึ่งเป็นการค้นหาในบริเวณเดิมซ้ำๆ จึงทำให้ผลลัพธ์ที่ได้ไม่ดีเท่าที่ควร ดังรูปที่ 4.17



รูปที่ 4.17 สาทิตของ RN

รูป 4.17 คือภาพสาทิตของ RN อนุภาคถูกดำเนินการรีโพอิชัน ทำให้อนุภาคกระจายตัวออกจากจุดสุดสัมพัทธ์ตามลูกศรเส้นประสีน้ำเงิน แต่เนื่องจาก PBEST คือวงกลมเส้นประสีส้ม และ GBEST คืออนกที่สวมมงกุฎอยู่ที่ตำแหน่งจุดสุดสัมพัทธ์เดิม เมื่อดำเนินการค้นหาด้วย PSO อนุภาคจะลู่กลับเข้ามา GBEST ซึ่งอยู่ในจุดสุดสัมพัทธ์เดิมตามลูกศรเส้นประสีแดง

เพื่อหลีกเลี่ยงเหตุการณ์กลับเข้าไปค้นหาในบริเวณเดิม รีโพอิชชันควรถูกรีเซ็ตทั้ง GBEST และ PBEST หรือ RPG เพื่อให้การรีเซ็ตทำให้อนุภาคมีโอกาสเข้าไปค้นหาในบริเวณใหม่ ซึ่งบริเวณใหม่นี้อาจจะเป็นตำแหน่งที่ดีกว่าหรือแย่กว่าตำแหน่งก่อนที่ถูกรีเซ็ตดังรูป 4.18 โดยจากกราฟ 4.16 รอบรีเซ็ตที่เริ่มต้นที่จุด C แสดงให้เห็นว่าอนุภาคเข้าไปค้นหาบริเวณใหม่ เพราะระดับเส้นกราฟที่ติดจุดสุดสัมพัทธ์อยู่ในระดับที่แตกต่างจากระดับเส้นกราฟก่อนรีเซ็ต ซึ่งแตกต่างจาก RN, RP, RG ที่หลังจากการรีเซ็ตอนุภาคเข้าไปค้นหาในบริเวณเดิมก่อนรีเซ็ต และจากตารางผลการทดลองที่ 5.13 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 แสดงให้เห็นว่า RPG ได้รับคำตอบการค้นหาดีกว่า RN, RP, RG, PSO ซึ่งเป็นผลมาจากการที่อนุภาคมีโอกาสเข้าไปค้นหาในบริเวณใหม่ แต่ก็ไม่มีอะไรรับประกันว่าอนุภาคอาจจะเคลื่อนที่กลับไปยังตำแหน่งเดิมถึงแม้ว่าจะดำเนินการรีเซ็ต GBEST และ PBEST โดยจากกราฟ 4.16 รอบรีเซ็ตที่เริ่มต้นที่จุด D แสดงให้เห็นว่าอนุภาคเข้าไปค้นหาบริเวณเดิมก่อนถูกรีเซ็ตเพราะระดับเส้นกราฟที่ติดจุดสุดสัมพัทธ์อยู่ในระดับเท่ากับระดับเส้นกราฟก่อนถูกรีเซ็ต จากเหตุผลที่กล่าวมานี้สามารถสรุปได้ว่า RPG มีประสิทธิภาพการค้นหาคำตอบที่ดีที่สุดเมื่อเปรียบเทียบกับ RN, RP, RG, PSO



รูปที่ 4.18 สาทิตของ RPG

จากรูป 4.18 อนุภาคถูกดำเนินการรีโพอิชชัน ทำให้อนุภาคกระจายตัวออกจากจุดสุดสัมพัทธ์ตามลูกศรเส้นประสีน้ำเงิน สำหรับ PBEST คือวงกลมเส้นประสีส้มและ GBEST คือนกที่สวมมงกุฎ ซึ่งทั้ง PBEST และ GBEST ถูกรีเซ็ต ทำให้ GBEST เปลี่ยนตำแหน่งไปตามลูกศรเส้นประสีเขียว และ PBEST กระจายตัวออกจากจุดสุดสัมพัทธ์และอยู่ใกล้กับอนุภาคในตำแหน่งใหม่ เมื่อดำเนินการค้นหาด้วย PSO อนุภาคจะลู่กลับเข้ามา GBEST ตามลูกศรเส้นประสีแดง ซึ่งเป็นการลู่เข้าหาตำแหน่งบริเวณใหม่ทำให้การค้นหาที่นั่นเกิดการค้นหาคำตอบในบริเวณใหม่จึงมีโอกาสพบคำตอบที่ดีขึ้นกว่าเดิม

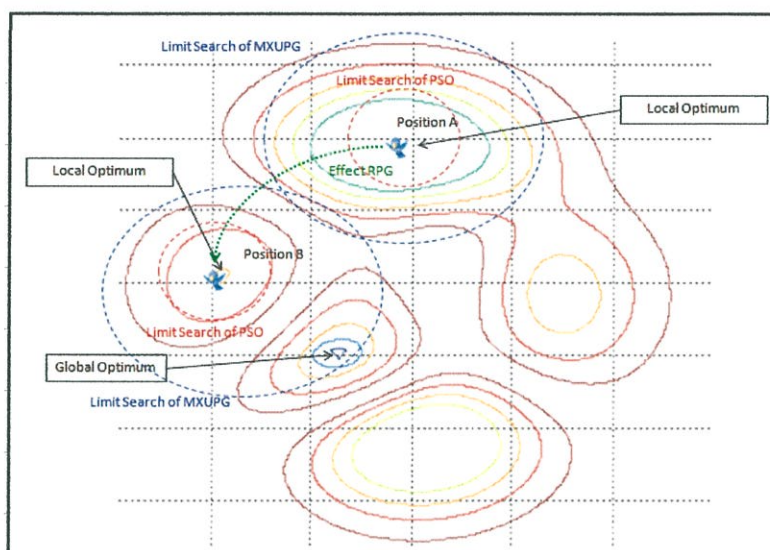
เนื่องจาก RPG มีโอกาสลู่เข้าไปค้นหาคำตอบในบริเวณใหม่ ซึ่งจะทำให้ได้ผลลัพธ์การค้นหาดีกว่า RN, RP, RG ที่ดำเนินการค้นหาโดยกลับเข้าไปค้นหาในบริเวณเดิมก่อนรีเซ็ตเท่านั้น ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้นำเสนอให้นำ RPG มาใช้ปรับปรุงการทำงานของ PSO

4.3 วิธีการรีโพซิชันสำหรับ PSO ที่ประยุกต์ใช้การกลายพันธุ์

การทดลองปัญหาฟังก์ชันมาตรฐานจากตารางผลการทดลองที่ 5.14 ซึ่งมาจากผลการทดลองในหัวข้อ 5.8 แสดงให้เห็นว่าบางปัญหา RPG จะได้ผลลัพธ์การค้นหาที่ดีกว่า MXUPG ในทางกลับกัน บางปัญหา MXUPG ก็จะได้ผลลัพธ์การค้นหาที่ดีกว่า RPG เช่นกัน สำหรับผลการทดลองการแก้ปัญหาถุงเป้หลายมิติจากตารางผลการทดลอง 5.15 ซึ่งมาจากผลการทดลองในหัวข้อ 5.9 พบว่า RPG ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า MXUPG เนื่องจากปัญหารูปแบบนี้เหมาะสมที่จะแก้ปัญหาด้วย RPG มากกว่า MXUPG แต่สำหรับผลการทดลองการแก้ปัญหาการเดินทางของพนักงานขายจากตารางผลการทดลองที่ 5.16 ซึ่งมาจากผลการทดลองในหัวข้อ 5.10 พบว่า MXUPG ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า RPG เนื่องจากปัญหารูปแบบนี้เหมาะสมที่จะแก้ปัญหาด้วย MXUPG มากกว่า RPG

จากผลการทดลองในหัวข้อ 5.8, 5.9, 5.10 แสดงให้เห็นว่า MXUPG และ RPG มีประสิทธิภาพที่จะแก้ไขปัญหาการหาจุดที่ดีที่สุดที่แตกต่างกัน เนื่องจากทั้งสองวิธีนั้นได้ประสบปัญหาที่แตกต่างกัน โดยปัญหาของ MXUPG คือปัญหาขอบเขตการกลายพันธุ์และสำหรับปัญหาของ RPG คือปัญหาขาดความหลากหลาย โดยรายละเอียดที่ RPG นั้นประสบปัญหาขาดความหลากหลายถูกกล่าวไว้ในภาคผนวก ฅ.4 สำหรับปัญหาของ MXUPG สามารถใช้ RPG แก้ปัญหาได้ ในขณะที่ปัญหาของ RPG สามารถใช้ MXUPG แก้ปัญหาได้ เนื่องจากทั้งสองวิธีนี้สามารถนำมาประยุกต์ใช้ร่วมกัน และสามารถเพิ่มประสิทธิภาพของการค้นหาของ PSO โดยสามารถลดข้อด้อยของกันและกัน ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้นำเสนอให้นำทั้งสองวิธีมาประยุกต์ใช้ร่วมกัน

โดย MXUPG จะเพิ่มความหลากหลายของประชากรเมื่ออนุภาคติดจุดสุดสัมพัทธ์ ทำให้เกิดการค้นหารอบจุดสุดสัมพัทธ์ ถ้าหากจุดสุดสัมบูรณ์อยู่รอบบริเวณนี้จะทำให้สามารถหาคำตอบพบ ดังรูป ฅ.17 หรือถ้าหากมีตำแหน่งที่ดีกว่าจุดสุดสัมบูรณ์อยู่รอบบริเวณที่อนุภาคติดจุดสุดสัมพัทธ์อยู่ MXUPG จะช่วยให้อนุภาคหลุดออกจากจุดสุดสัมพัทธ์ที่ติดอยู่ไปติดจุดสุดสัมพัทธ์ที่ดีกว่าแทน ดังนั้น MXUPG จึงสามารถแก้ปัญหาขาดความหลากหลายได้ ถ้าหากค้นหาด้วย MXUPG แล้วคำตอบไม่ได้เป็นเวลานานจนสามารถสรุปได้ว่า MXUPG เกิดปัญหาขอบเขตการกลายพันธุ์ เพื่อจัดการกับปัญหานี้จึงนำ RPG มาประยุกต์ใช้ ซึ่งจะช่วยให้อนุภาคการกระจายตัวออกจากจุดสุดสัมพัทธ์แล้วอนุภาคจะมีโอกาสลู่เข้าไปค้นหาในบริเวณใหม่ดังรูป 4.19 หลังจากนั้นจะดำเนินการค้นหาคำตอบด้วย MXUPG จนกว่าจะประสบปัญหาการติดในจุดสุดสัมพัทธ์อีกครั้งจึงจะดำเนินการแก้ปัญหาด้วย RPG อีก วนลูปการทำงานเช่นนี้ไปเรื่อยๆ จนกว่าจะพบคำตอบหรือรอบการค้นหาคำตอบถึงจำนวนที่กำหนด ซึ่งเรียกวิธีการที่นำเสนอว่า การประยุกต์การกลายพันธุ์และรีโพซิชันกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Applying Mutation and Reposition to Particle Swarm Optimization หรือ MRPSO)



รูปที่ 4.19 สาธิตการแก้ปัญหาขอบเขตการกลายพันธุ์

จากรูป 4.19 กรณีนี้อนุภาคติดอยู่ในจุดสุดสัมพัทธ์ที่ตำแหน่ง A โดยขอบเขตการค้นหาด้วย PSO จะอยู่ในวงกลมสีแดงเส้นประ สำหรับการค้นหาแบบ MXUPG ช่วยเพิ่มขอบเขตการค้นหาของ PSO ให้กว้างขึ้น โดยมีขอบเขตการค้นหาอยู่ในวงกลมสีน้ำเงินเส้นประ ซึ่ง MXUPG ไม่สามารถหาจุดสุดสัมบูรณ์เพราะในขอบเขตการกลายพันธุ์ไม่มีจุดสุดสัมบูรณ์อยู่ เมื่อ RPG ดำเนินการจะทำให้อนุภาคลู่เข้าหาจุดสุดสัมพัทธ์ในตำแหน่งใหม่ที่ตำแหน่ง B ซึ่งในจุดสุดสัมพัทธ์นี้ ขอบเขตการกลายพันธุ์สามารถครอบคลุมจุดสุดสัมบูรณ์การค้นหาจึงมีโอกาสพบคำตอบ

โดยขั้นตอนการทำงานของ MRPSO สามารถสรุปหลักการได้ดังนี้ โดยเริ่มต้นใช้ MXUPG ค้นหาคำตอบซึ่งจะทำให้อนุภาคต่างๆ ลู่เข้าหาจุดสุดสัมพัทธ์หรือไม่ก็อาจจะพบคำตอบถ้าไม่เกิดปัญหาขอบเขตการกลายพันธุ์ขึ้น แต่ถ้า MXUPG เกิดปัญหาขอบเขตการกลายพันธุ์คือเมื่อดำเนินการกลายพันธุ์เป็นเวลานานแล้วคำตอบที่ได้ยังไม่ดีขึ้น แสดงว่าบริเวณโดยรอบไม่มีตำแหน่งที่ดีกว่าเดิมแล้ว จึงดำเนินขั้นตอน RPG เพื่อจัดการกับปัญหาขอบเขตการกลายพันธุ์ ซึ่งทำให้เกิดการกระจายอนุภาคออกมาในบริเวณที่กว้างกว่าการกลายพันธุ์ปกติ หลังจากนั้นจะดำเนินการค้นหาด้วย MXUPG อีกครั้ง จนพบปัญหาขอบเขตการกลายพันธุ์อีกครั้งก็จะดำเนินขั้นตอน RPG ซึ่งการค้นหาจะวนซ้ำเช่นนี้ไปเรื่อยๆ จนกว่าจะพบคำตอบหรือรอบการค้นหาคำตอบถึงจำนวนที่กำหนด โดยรหัสจำลองของ MRPSO แสดงดังรูป 4.20

1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)	1
2) กำหนดค่า PM, RM, PR, TR	2
3) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)	3
4) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)	4
5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้นๆ ($PBEST_i = \vec{x}_i$)	5
6) ปรับปรุงความเร็วของแต่ละอนุภาคและตำแหน่งของแต่ละอนุภาคตามสมการ 2.2 และ 2.3	6
7) วนซ้ำจาก i = 1 ถึง N	(Mutation step) 7
วนซ้ำจาก j = 1 ถึง RM	8
วนซ้ำจาก d = 1 ถึง D	9
$Tx_d = x_{id}$	10
ถ้า $PM > rand()$: ดำเนินตามสมการ 4.1	11
ถ้า fitness ของ Tx ดีกว่า fitness ของ PBEST: $PBEST = Tx$	12
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST: $GBEST = Tx$	(End mutation step) 13
8) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$	(Reposition step) 14
รีเซ็ต GBEST, PBEST	15
วนซ้ำจาก i = 1 ถึง N	16
วนซ้ำจาก d = 1 ถึง D	17
ถ้า $PR > rand()$: ดำเนินตามสมการ 4.2	(End reposition step) 18
9) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$	19
หยุดการดำเนินการค้นหาคำตอบ	20
นอกจากนี้	21
วนซ้ำไปยังขั้นตอนที่ 3)	22

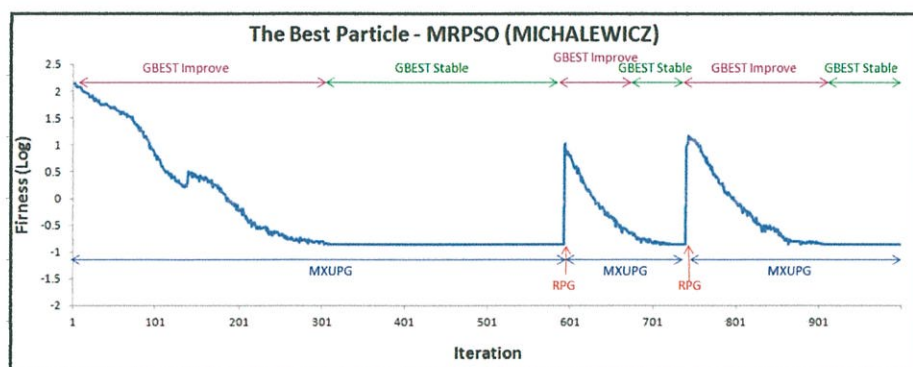
รูปที่ 4.20 รหัสจำลองของ MRPSO

โดยสมการ 4.1 และ 4.2 คือสมการกลายพันธ์ที่ถูกใช้กับ MRPSO โดยถ้าการสุ่มค่าในช่วง 0 ถึง 1 มีค่ามากกว่า 0.5 จะเลือกใช้ค่าบวกในสมการ แต่ถ้าการสุ่มค่าในช่วง 0 ถึง 1 มีค่าน้อยกว่าหรือเท่ากับ 0.5 จะเลือกใช้ค่าลบในสมการ

$$Tx_d = Tx_d \pm (Tx_d \times rand ()) \quad (4.1)$$

$$x_{id} = x_{id} \pm (x_{id} \times rand ()) \quad (4.2)$$

โดยที่ RM หรือ the round of mutation คือรอบที่จะดำเนินการกลายพันธ์ PM หรือ the probability of mutation คือความน่าจะเป็นในการกลายพันธ์ PR หรือ the probability of reposition คือความน่าจะเป็นในการรีโพซิชั่น TR หรือ the threshold of reposition คือจำนวนซ้ำของ GBEST อย่างต่อเนื่องที่รีโพซิชั่นจะถูกดำเนินการ Tx หรือ the temporary particle คืออนุภาคชั่วคราว Tx_d คืออนุภาคชั่วคราวของมิติ d x_{id} คือตำแหน่งของอนุภาค i และมิติ d N คือจำนวนของประชากร และ D คือจำนวนมิติของปริภูมิของคำตอบ



รูปที่ 4.21 ขั้นตอนการทำงานของ MRPSO

จากกราฟ 4.21 ช่วงแรกของการค้นหา GBEST จะมีค่าลดลงอย่างรวดเร็วเป็นผลเนื่องจากการค้นหาแบบ MXUPG และเป็นช่วงที่ไม่เกิดปัญหาขอบเขตการกลายพันธุ์ในช่วงต่อมาเกิดปัญหาขอบเขตการกลายพันธุ์ค่าของ GBEST จะไม่สามารถปรับปรุงค่าตอบได้เป็นเวลานานจนครบรอบที่กำหนดจึงดำเนินการ RPG หลังจากดำเนินการ RPG ก็จะมีการค้นหาด้วย MXUPG ใหม่อีกครั้ง และจะวนซ้ำเช่นนี้ไปเรื่อยๆ จนพบคำตอบหรือครบรอบการค้นหาคำตอบที่ได้กำหนดเอาไว้

4.4 วิเคราะห์ข้อดีและข้อเสียของวิธีที่นำเสนอ

4.4.1 ข้อดีของวิธีที่นำเสนอ

วิธี MRPSO เป็นวิธีที่พัฒนาจาก PSO จึงได้รับข้อดีของ PSO มาด้วย นอกจากนี้วิธีนี้ได้ลดปัญหาการติดในจุดสุดสัมพัทธ์ ปัญหาการปรับพารามิเตอร์ของการกลายพันธุ์ ปัญหาขอบเขตการกลายพันธุ์ และปัญหาขาดความหลากหลาย จึงทำให้คำตอบจากการค้นหานี้ดีกว่า PSO โดยข้อดีของวิธีนี้มีดังนี้

- สามารถนำไปประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดที่คำตอบสามารถแทนด้วยจุดบนปริภูมิขนาดหลายมิติ โดยคำตอบจะถูกนำมาแทนให้อยู่ในรูปแบบของอนุภาค
- สามารถประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดที่มีปริภูมิของคำตอบขนาดใหญ่มาก
- มีความเร็วในการลู่เข้าหาคำตอบมาก
- สามารถเขียนโปรแกรมหรือนำมาประยุกต์ใช้ได้ง่าย และหลังจากพัฒนาเสร็จสามารถประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดในรูปแบบอื่นได้ง่าย
- มีจำนวนพารามิเตอร์เพียงเล็กน้อยที่จำเป็นต้องกำหนดเพื่อใช้ในการค้นหาคำตอบ
- เป็นขั้นตอนวิธีที่ง่ายไม่ซับซ้อน
- สามารถลดปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO เพราะนำ MXUPG และ RPG มาประยุกต์ใช้จึงได้รับคำตอบจากการค้นหาที่ดีกว่า PSO

- สามารถนำมาประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดทั่วไปได้ง่ายขึ้น เมื่อเปรียบเทียบกับกรการกลายพันธุ์แบบอื่น เนื่องจากไม่ประสบปัญหาการปรับพารามิเตอร์ จึงสามารถปรับพารามิเตอร์ประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดทั่วไปได้ง่ายขึ้น
- MRPSO สามารถลดหรือแก้ไขปัญหาคอขวดของการกลายพันธุ์ เพราะมีขั้นตอน RPG จึงได้รับคำตอบจากการค้นหาดีกว่า PSO
- MRPSO สามารถลดหรือแก้ไขปัญหาคอขวดของความหลากหลาย เพราะมีขั้นตอน MXUPG จึงได้รับคำตอบจากการค้นหาดีกว่า PSO
- สำหรับในกรณีศึกษาของวิทยานิพนธ์ฉบับนี้ ได้แก่ ฟังก์ชันมาตรฐาน ปัญหาถูกเปลี่ยนมิติ ปัญหาการเดินทางของพนักงานขาย MRPSO ได้รับผลลัพธ์จากการค้นหาดีกว่า PSO ทั้งในด้านคำตอบที่ได้จากการค้นหาและความน่าเชื่อถือของขั้นตอนวิธี ดังนั้นเมื่อนำ MRPSO ไปประยุกต์กับปัญหาที่มีรูปแบบเหล่านี้ MRPSO จะได้รับคำตอบที่ดีกว่า PSO

4.4.2 ข้อเสียของวิธีที่นำเสนอ

เนื่องจากวิธี MRPSO เป็นวิธีที่พัฒนาจาก PSO ปัญหาของ MRPSO จึงยังคงประสบปัญหาเดียวกับ PSO คือปัญหาการติดในจุดสุดสัมพัทธ์ นอกจากนี้ MRPSO ต้องใช้เวลาในการค้นหาคำตอบนานมากเพื่อให้ได้คำตอบเทียบเท่ากับขั้นตอนวิธีที่แก้ปัญหามาโดยเฉพาเจาะจง และสุดท้าย MRPSO ได้เพิ่มขั้นตอนและพารามิเตอร์จาก PSO ปกติ ส่งผลให้เวลาที่ใช้และความยุ่งยากของการกำหนดค่าพารามิเตอร์เพิ่มขึ้น โดยข้อเสียของ MRPSO มีดังนี้

- MRPSO ยังคงประสบปัญหาการติดในจุดสุดสัมพัทธ์ ถึงแม้ MRPSO จะมี MXUPG ที่ทำให้ขอบเขตในการค้นหาวางกว้างกว่า PSO เมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ แต่ก็มีขอบเขตที่จำกัด และ RPG ที่ทำให้เกิดการกระจายตัวไปค้นหาในบริเวณอื่น แต่ RPG ไม่รับประกันว่าเกิดการค้นหาบริเวณใหม่ทุกครั้ง ดังนั้นการค้นหาก็จะค้นหาบริเวณเก่าที่เคยค้นหาแล้วซ้ำไปซ้ำมาจึงไม่สามารถแก้ปัญหาคอขวดติดในจุดสุดสัมพัทธ์ได้สมบูรณ์แบบ
- ไม่ทราบว่าปัญหาที่แก้ไขนั้นต้องกำหนดจำนวนอนุภาคและจำนวนรอบการค้นหาเท่าไรจึงจะเพียงพอที่จะแก้ปัญหานั้น เพราะถ้าหากกำหนดจำนวนอนุภาคหรือจำนวนรอบการค้นหามากเกินไปก็จะสูญเสียเวลาในการค้นหา และที่สำคัญเมื่อนำมาประยุกต์ใช้งานจริงเราไม่ทราบคำตอบที่ดีที่สุดของปัญหาที่แก้ไข MRPSO อาจจะสามารถหาคำตอบที่ดีที่สุดพบตั้งนานแล้ว แต่การค้นหา ยังคงไม่หยุดจึงเสียเวลาการค้นหาโดยเปล่าประโยชน์ ในทางกลับกันถ้ากำหนดจำนวนอนุภาคหรือจำนวนรอบการค้นหาน้อยเกินไป MRPSO ก็ไม่สามารถหาคำตอบที่ดีที่สุดพบ ซึ่งในการใช้งานจริง เราทราบไม่ว่าคำตอบที่ได้จากการค้นหานั้นเป็นคำตอบที่ดีที่สุดหรือยัง

- การรีเซ็ตของ MRPSO ในบางครั้งอาจจะไม่มีประโยชน์ เพราะการรีเซ็ตก็ไม่รับประกันว่าอนุภาคได้เคลื่อนที่ไปค้นหาในบริเวณใหม่ทุกครั้ง เนื่องจากการรีเซ็ตเป็นการทำงานแบบสุ่มจึงมีโอกาสที่จะค้นหาในบริเวณใหม่ และก็มีโอกาสกลับมาค้นหาในบริเวณเก่า ซึ่งถ้าหากกลับมาค้นหาในบริเวณเก่าการค้นหาในรอบรีเซ็ตนั้นก็ไม่มีประโยชน์
- MRPSO ต้องใช้เวลาการค้นหาเพิ่มขึ้น เนื่องจาก MRPSO ได้นำ MXUPG และ RPG เพิ่มเข้าไปในขั้นตอนการทำงานของ PSO เมื่อเปรียบกับ PSO เวลาในการคำนวณในแต่ละรอบการค้นหาจะเพิ่มขึ้น
- การเขียนโปรแกรมประยุกต์ใช้กับปัญหาทั่วไปนั้นยุ่งยากกว่า PSO เพราะต้องเพิ่มขั้นตอน MXUPG และ RPG
- เมื่อเปรียบเทียบกับขั้นตอนวิธีที่แก้ปัญหาโดยเฉพาะเจาะจง MRPSO ต้องใช้เวลาการคำนวณเป็นหลายเท่าจึงจะได้คำตอบเทียบเท่ากับขั้นตอนวิธีที่แก้ปัญหาโดยเฉพาะเจาะจง เพราะ MRPSO ใช้วิธีการค้นหาแบบสุ่มจึงต้องในเวลามากกว่าจะปรับปรุงคำตอบให้ดีขึ้นเทียบเท่ากับขั้นตอนวิธีที่แก้ปัญหาโดยเฉพาะเจาะจง
- MRPSO ใช้วิธีการค้นหาแบบสุ่ม ดังนั้นการค้นหาแต่ละครั้งอาจจะได้คำตอบไม่เท่ากัน และไม่มีการรับประกันคำตอบที่ได้นั้นจะเป็นคำตอบที่ดีที่สุดของปัญหาที่แก้ไข ซึ่งแตกต่างจากขั้นตอนวิธีที่แก้ปัญหาโดยเฉพาะเจาะจง คือเมื่อหาคำตอบพบไม่ว่าจะค้นหาใหม่กี่ครั้งคำตอบที่ได้ก็จะยังคงเป็นคำตอบเดิม และสามารถรับประกันคำตอบที่ได้เป็นคำตอบที่ดีที่สุดของปัญหาที่แก้ไข
- การนำไปประยุกต์ใช้กับปัญหาทั่วไปนั้นยุ่งยากกว่า PSO เพราะต้องวิเคราะห์พารามิเตอร์ที่เพิ่มขึ้นจาก PSO ให้เหมาะสมกับปัญหาที่แก้ไข ไม่เช่นนั้นคำตอบที่ได้จะแยกว่า PSO หรือไม่มีประสิทธิภาพเท่าที่ควร โดยมีจำนวนพารามิเตอร์เพิ่มขึ้นคือ Round of Mutation (RM), Probability of Mutation (PM), Probability of Reposition (PR) และ Threshold of Reposition (TR) ถ้ากำหนดพารามิเตอร์เหล่านี้ไม่เหมาะสมกับปัญหาที่แก้ไข ผลลัพธ์จากการค้นหาจะไม่มีประสิทธิภาพเท่าที่ควร เช่น RM และ PM ถ้ากำหนดไม่เหมาะสม อาจจะได้คำตอบใกล้เคียงกับ PSO ที่มีจำนวนอนุภาคเท่ากัน เทคนิคการกลายพันธุ์ที่เพิ่มเข้าไบนั้นไม่ทำให้คำตอบดีขึ้น สำหรับ PR และ TR ถ้ากำหนดให้ดำเนินการมากเกินไปจะทำให้เกิดปัญหาการกลายพันธุ์มากเกินไป ผลลัพธ์ที่ได้จากการค้นหาจะแยกว่า PSO ในทางกลับกัน ถ้ากำหนดให้ดำเนินการน้อยเกินไปการรีเซ็ตก็จะไม่สามารถปรับปรุงคำตอบ เทคนิคที่เพิ่มเข้าไปก็จะไม่มีประโยชน์ แต่โดยส่วนมากพารามิเตอร์สองตัวนี้กำหนดได้ง่ายคือให้มีความถี่สูงไว้ก่อนเพื่อป้องกันปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

บทที่ 5

การทดลองการประยุกต์การกลายพันธุ์และรีโพอิชัน

ในบทนี้จะกล่าวถึงการทดลองเพื่อเปรียบเทียบผลลัพธ์จากการค้นหาของปัญหาฟังก์ชันมาตรฐาน (Benchmark Test Functions) ปัญหาถุงเป้หลายมิติ (Multidimensional Knapsack Problems หรือ MKP) และปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem หรือ TSP) โดยใช้วิธีการประยุกต์การกลายพันธุ์และรีโพอิชันกับ PSO โดยมีรายละเอียดวิธีต่างๆ ดังนี้

- วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization หรือ PSO) [1, 2]
- วิธีการกลายพันธุ์แบบรอบการค้นหาคำตอบที่คงที่ (Fix Iteration Mutation หรือ FIM) [17] (ซึ่งเป็นขั้นตอนวิธีที่ใช้แทนวิธีประยุกต์การกลายพันธุ์ประเภท NCM)
- MGUG [34-42] คือวิธีประยุกต์การกลายพันธุ์ประเภท CM โดยดำเนินการกลายพันธุ์รอบ GBEST แล้วนำผลการกลายพันธุ์มาปรับปรุง GBEST (ซึ่งเป็นวิธีที่นำมาช่วยวิเคราะห์และเปรียบเทียบผลการทดลองกับวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ)
- MPUP [45, 46] คือวิธีประยุกต์การกลายพันธุ์ประเภท CM โดยดำเนินการกลายพันธุ์รอบ PBEST แล้วนำผลการกลายพันธุ์มาปรับปรุง PBEST และ GBEST (ซึ่งเป็นวิธีที่นำมาช่วยวิเคราะห์และเปรียบเทียบผลการทดลองกับวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ)
- MXUX [43, 44] คือวิธีประยุกต์การกลายพันธุ์ประเภท CM โดยดำเนินการกลายพันธุ์รอบตำแหน่งของอนุภาค แล้วนำผลการกลายพันธุ์มาปรับปรุงตำแหน่งของอนุภาค PBEST และ GBEST (ซึ่งเป็นวิธีที่นำมาช่วยวิเคราะห์และเปรียบเทียบผลการทดลองกับวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ)
- MXUPG [84] คือวิธีประยุกต์การกลายพันธุ์ประเภท CM โดยดำเนินการกลายพันธุ์รอบตำแหน่งของอนุภาค แล้วนำผลการกลายพันธุ์มาปรับปรุง PBEST และ GBEST เท่านั้น (ซึ่งเป็นส่วนประกอบของวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ)
- รีรัน หรือ RERUN [48-52] คือวิธีการค้นหาคำตอบด้วยรีรัน โดยรีเซ็ตตำแหน่งของอนุภาคทั้งหมดเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยปราศจากการรีเซ็ต PBEST และ GBEST จากนั้นจึงดำเนินการค้นหาด้วย PSO
- Reposition Non-Reset หรือ RN คือวิธีการค้นหาคำตอบด้วยรีโพอิชัน โดยรีเซ็ตตำแหน่งของอนุภาคเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์โดยปราศจากการรีเซ็ต PBEST และ GBEST จากนั้นจึงดำเนินการค้นหาด้วย PSO (ซึ่งเป็นวิธีที่นำมาช่วยวิเคราะห์และเปรียบเทียบผลการทดลองกับวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ)

- Reposition with GBEST Reset หรือ RG คือวิธีการค้นหาคำตอบด้วยวิธีโพซิชั่น โดยรีเซ็ตตำแหน่งของอนุภาคเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์โดยปราศจากการรีเซ็ต PBEST แต่ GBEST ถูกรีเซ็ตจากนั้นจึงดำเนินการค้นหาด้วย PSO (ซึ่งเป็นวิธีที่นำมาช่วยวิเคราะห์และเปรียบเทียบผลการทดลองกับวิธีที่วิทยานิพนธ์ฉบับนี้ นำเสนอ)
- Reposition with PBEST Reset หรือ RP คือวิธีการค้นหาคำตอบด้วยวิธีโพซิชั่น โดยรีเซ็ตตำแหน่งของอนุภาคเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์โดยปราศจากการรีเซ็ต GBEST แต่ PBEST ถูกรีเซ็ตจากนั้นจึงดำเนินการค้นหาด้วย PSO (ซึ่งเป็นวิธีที่นำมาช่วยวิเคราะห์และเปรียบเทียบผลการทดลองกับวิธีที่วิทยานิพนธ์ฉบับนี้ นำเสนอ)
- Reposition with PBEST and GBEST Reset หรือ RPG [84] คือวิธีการค้นหาคำตอบด้วยวิธีโพซิชั่น โดยรีเซ็ตตำแหน่งของอนุภาคเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์และรีเซ็ต GBEST และ PBEST จากนั้นจึงดำเนินการค้นหาด้วย PSO (ซึ่งเป็นส่วนประกอบของวิธีที่วิทยานิพนธ์ฉบับนี้ นำเสนอ)
- การประยุกต์การกลายพันธุ์และวิธีโพซิชั่นกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Applying Mutation and Reposition to Particle Swarm Optimization หรือ MRPSO) (ซึ่งเป็นวิธีที่วิทยานิพนธ์ฉบับนี้ นำเสนอ) [84]
- การประยุกต์ใช้วิธีการกลายพันธุ์โดยการปรับพารามิเตอร์กับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (an Adaptive Particle Swarm Optimization หรือ APSO) [33]
- การประยุกต์ใช้วิธีการกลายพันธุ์แบบลูกผสมกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (A Hybrid Particle Swarm Optimization which incorporates mutation operation หรือ HPSO) [34]
- การประยุกต์ใช้วิธีรีรันโดยพิจารณาจากระยะทางยูคลิดกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO with Rerun by considering from the Euclidean distance หรือ PSORE) [51]
- การประยุกต์ใช้วิธีรีรันโดยพิจารณาจากตำแหน่งที่ดีที่สุดของอนุภาคข้างเคียงกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO with Rerun by considering from the neighborhood's best position หรือ PSORN) [48]
- วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในรูปแบบไบนารี (Binary Particle Swarm Optimization หรือ BPSO) [102]
- การประยุกต์ใช้หลักการจีโนไทป์-ฟีโนไทป์กับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในรูปแบบไบนารี (Applying Genotype–Phenotype to Binary Particle Swarm Optimization หรือ GPBPSO) [103]
- การดัดแปลงวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในรูปแบบไบนารี (Modified Binary Particle Swarm Optimization หรือ MBPSO) [104]

- วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในรูปแบบการสลับลำดับ (Permutation Particle Swarm Optimization หรือ PPSO) [105]
- วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในรูปแบบผสมของการสลับลำดับ (A Mixed Permutation Particle Swarm Optimization หรือ MPPSO) [106]

การทดลองนี้ได้ทดลองกับปัญหาทั้งหมด 3 ปัญหา โดยมีรายละเอียดดังนี้

- การทดลองกับฟังก์ชันมาตรฐาน 24 ฟังก์ชัน [54 - 60, 100, 101, 113] จากตาราง ก.1 โดยที่ 16 ฟังก์ชันแรกเป็นปัญหามัลติโมดอล และสำหรับ 8 ฟังก์ชันหลังเป็นปัญหายูนิโมดอล จุดประสงค์ของการทดลองชุดนี้ คือวัดประสิทธิภาพการค้นหาของขั้นตอนวิธีที่นำเสนอเปรียบเทียบกับ PSO, APSO, HPSO, PSORE และ PSORN โดยฟังก์ชันมาตรฐานเป็นปัญหาที่มีรูปแบบปริภูมิคำตอบถูกแทนด้วยจำนวนจริง นอกจากนี้การทดลองนี้ถูกใช้อธิบายแนวคิดที่มาของขั้นตอนวิธีที่นำเสนอ
- การทดลองกับปัญหาถุงเป่าหลายมิติ โดยปัญหานี้ได้นำมาจากชุดการทดลองของ OR-Library [61] ทั้งหมด 5 ชุดดังนี้ SENTO [107], WEING [108], WEISH [109], PB [110] และ HP [111] โดยทั้งหมดมีจำนวนของตัวอย่าง 48 ตัวอย่าง จุดประสงค์ของการทดลองชุดนี้คือ วัดประสิทธิภาพการค้นหาของขั้นตอนวิธีที่นำเสนอเปรียบเทียบกับ BPSO, GPBPSO และ MBPSO โดยปัญหาถุงเป่าหลายมิติเป็นปัญหาที่มีรูปแบบปริภูมิคำตอบถูกแทนด้วยไบนารี
- การทดลองกับปัญหาการเดินทางของพนักงานขาย โดยปัญหานี้ได้นำมาจากชุดการทดลองของ TSPLIB [62] โดยตัวอย่างที่นำมาทดลองมีทั้งหมด 14 ตัวอย่าง จุดประสงค์ของการทดลองชุดนี้คือ วัดประสิทธิภาพการค้นหาของขั้นตอนวิธีที่นำเสนอเปรียบเทียบกับ PPSO และ MPPSO โดยปัญหาการเดินทางของพนักงานขายเป็นปัญหาที่มีรูปแบบปริภูมิคำตอบถูกแทนด้วยการสลับลำดับกันในพีชคณิต

การทดลองสามารถแบ่งออกเป็น 10 ส่วน โดยพิจารณาจากจุดประสงค์การทดลองมีรายละเอียดดังนี้

- การทดลองปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO โดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองการประยุกต์ใช้การกลายพันธุ์กับ PSO โดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองปัญหาจากการกลายพันธุ์แบบ NCM โดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองปัญหาจากการกลายพันธุ์และแนวทางการแก้ปัญหาการกลายพันธุ์แบบ CM โดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองการประยุกต์ใช้รีรันและรีโพซิชั่นโดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพซิชั่นโดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองผลกระทบจากการกลายพันธุ์และรีโพซิชั่นโดยทดลองกับฟังก์ชันมาตรฐาน

- การทดลองเปรียบเทียบประสิทธิภาพของการค้นหาของขั้นตอนวิธีที่นำเสนอกับ PSO ของงานวิจัยอื่นโดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองเปรียบเทียบประสิทธิภาพของการค้นหาของขั้นตอนวิธีที่นำเสนอกับ PSO ของงานวิจัยอื่นโดยทดลองกับปัญหาถุงเป้หลายมิติ
- การทดลองเปรียบเทียบประสิทธิภาพของการค้นหาของขั้นตอนวิธีที่นำเสนอกับ PSO ของงานวิจัยอื่นโดยทดลองกับปัญหาการเดินทางของพนักงานขาย

ตารางที่ 5.1 พารามิเตอร์ที่ใช้ทดลองกับฟังก์ชันคณิตศาสตร์

พารามิเตอร์	ค่า
จำนวนอนุภาค (กรณีไม่ดำเนินการ CM)	220 อนุภาค
จำนวนอนุภาค (กรณีดำเนินการ CM)	20 อนุภาค
จำนวนรอบการค้นหาค่าตอบสูงสุด (Max Iteration)	ตามตารางในภาคผนวก ก.1
ขอบเขตการค้นหาของแต่ละฟังก์ชัน	ตามตารางในภาคผนวก ก.1
จำนวนมิติของแต่ละฟังก์ชัน	ตามตารางในภาคผนวก ก.1
ความเร็วมากที่สุด (Max velocity)	ตามตารางในภาคผนวก ก.1
ความเร็วน้อยสุด (Min velocity)	ตามตารางในภาคผนวก ก.1
จำนวนครั้งในการทดลอง	100 ครั้งต่อฟังก์ชัน
ค่าความเฉื่อย (Weight)	0.72984 [50]
ค่าคงที่ของ PBEST (Constant number PBEST)	1.49618 [50]
ค่าคงที่ของ GBEST (Constant number GBEST)	1.49618 [50]
รอบของการกลายพันธุ์ของขั้นตอนวิธี CM (Round of Mutation)	10
ความน่าจะเป็นในการกลายพันธุ์ของขั้นตอนวิธี CM (Probability of Mutation)	0.05
รอบของการกลายพันธุ์ของขั้นตอนวิธี FIM (Round of Mutation)	20
ความน่าจะเป็นในการกลายพันธุ์ของขั้นตอนวิธี FIM (Probability of Mutation)	0.10
ค่าที่กำหนดรอบของรีโพสิชัน (Threshold of Reposition)	100
ความน่าจะเป็นในการรีโพสิชัน (Probability of Reposition)	0.30
ค่าคงที่ α ของขั้นตอนวิธี APSO	0.60[33]
ค่าคงที่ k ของขั้นตอนวิธี HPSO	10[34]
ค่าคงที่ m ของขั้นตอนวิธี HPSO	10[34]
ค่าคงที่ Mutation Probability ของขั้นตอนวิธี PSORE	0.04 [51]
ค่าคงที่ MAXSTEP ของขั้นตอนวิธี PSORE	10 [51]
ค่าที่ใช้หารสมการของ BORDERRADIUS ของขั้นตอนวิธี PSORE	1000 [51]
ค่าคงที่ MAXSTEP ของขั้นตอนวิธี PSORN	15 [48]
ระยะห่างของอนุภาคข้างเคียง (neighborhood length) ของขั้นตอนวิธี PSORN	2 [48]

ตารางที่ 5.2 พารามิเตอร์ที่ใช้ทดลองกับปัญหาถุงเป้หลายมิติ

พารามิเตอร์	ค่า
จำนวนอนุภาค (กรณีไม่ดำเนินการ CM)	1000 อนุภาค
จำนวนอนุภาค (กรณีดำเนินการ CM)	500 อนุภาค
จำนวนรอบการค้นหาค่าตอบสูงสุด (Max Iteration)	5000 รอบ
จำนวนครั้งในการทดลอง	100 ครั้งต่อปัญหา
ความน่าจะเป็นเริ่มต้นของการเลือก	0.5
ค่าน้ำหนัก (Weight)	0.689343 [104]
ค่าคงที่ของ PBEST (Constant number PBEST)	2.00 [104]
ค่าคงที่ของ GBEST (Constant number GBEST)	2.00 [104]
ค่าคงที่ของ λ	1.0 [104]
ความเร็วมากที่สุด (Max velocity)	4 [104]
ความเร็วน้อยสุด (Min velocity)	-4 [104]
รอบของการกลายพันธุ์ของขั้นตอนวิธี CM (Round of Mutation)	1
ความน่าจะเป็นในการกลายพันธุ์ของขั้นตอนวิธี CM (Probability of Mutation)	0.05
รอบของการกลายพันธุ์ของขั้นตอนวิธี FIM (Round of Mutation)	50
ความน่าจะเป็นในการกลายพันธุ์ของขั้นตอนวิธี FIM (Probability of Mutation)	0.005
ค่าที่กำหนดรอบของรีโพซิชั่น (Threshold of Reposition)	100
ความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition)	0.30

ตารางที่ 5.3 พารามิเตอร์ที่ใช้ทดลองกับปัญหาการเดินทางของพนักงานขาย

พารามิเตอร์	ค่า
จำนวนอนุภาค	ตามตารางในภาคผนวก จ.1
จำนวนรอบการค้นหาค่าตอบสูงสุด (Max Iteration)	50000
ความเร็วมากที่สุด (Max velocity)	100
จำนวนครั้งในการทดลอง	ตามตารางในภาคผนวก จ.1
ค่าน้ำหนัก (Weight)	0.90
ค่าคงที่ของ PBEST (Constant number PBEST)	0.50
ค่าคงที่ของ GBEST (Constant number GBEST)	0.10
รอบของการกลายพันธุ์ของขั้นตอนวิธี CM (Round of Mutation)	ตามตารางในภาคผนวก จ.1
ความน่าจะเป็นในการกลายพันธุ์ของขั้นตอนวิธี CM (Probability of Mutation)	0.05
รอบของการกลายพันธุ์ของขั้นตอนวิธี FIM (Round of Mutation)	100
ความน่าจะเป็นในการกลายพันธุ์ของขั้นตอนวิธี FIM (Probability of Mutation)	0.20
ค่าที่กำหนดรอบของรีโพซิชั่น (Threshold of Reposition)	100
ความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition)	0.60

โปรแกรมที่ใช้ในการทดลอง คือ Microsoft Visual Studio 2005 พัฒนาโปรแกรมโดยใช้ภาษา C++ ทำการทดลองบนเครื่องคอมพิวเตอร์ส่วนบุคคล มีหน่วยประมวลผลกลางยี่ห้อ Intel รุ่น Core i7 3770 ความเร็ว 2.4 GHz และหน่วยความจำหลักขนาด 8 GB

จากตาราง 5.1, 5.2 และ 5.3 จำนวนอนุภาคของขั้นตอนวิธีที่มีการประยุกต์ใช้ CM (MGUG, MPUP, MXUX, MXUPG, MRPSO) มีจำนวนอนุภาคน้อยกว่าวิธีอื่น เพราะ CM มีจำนวนในการเดาคำตอบที่มาจากกรกลายพันธุ์ ดังนั้นเพื่อความยุติธรรมจำนวนเดาคำตอบที่มาจากกรกลายพันธุ์ จะถูกทดแทนด้วยจำนวนอนุภาค

สมการที่ใช้ดำเนินการกลายพันธุ์ของขั้นตอนวิธีทั้งหมดจะใช้สมการชุดเดียวกัน คือสมการ 4.1 ใช้สำหรับการกลายพันธุ์แบบ CM (MXUX, MPUP, MGUG, MXUPG, MRPSO, HPSO) และสมการ 4.2 ใช้สำหรับการกลายพันธุ์แบบ NCM (FIM, APSO) และรีโพลิชัน (RN, RP, RG, RPG) เพื่อให้เกิดความยุติธรรมในการวัดประสิทธิภาพของขั้นตอนวิธีสำหรับการแก้ปัญหาฟังก์ชันมาตรฐาน

สำหรับรหัสเทียม (Pseudo Code) ของขั้นตอนวิธี FIM, MGUG, MPUP, MXUX, MXUPG, RERUN, RN, RG, RP และ RPG จะถูกกล่าวในภาคผนวก ฉ สำหรับรหัสเทียมของขั้นตอนวิธีของ APSO, HPSO, PSORE, PSORN, BPSO, GPBPSO, MBPSO, PPSO และ MPPSO จะถูกกล่าวไว้เอกสารต้นฉบับ [33, 34, 51, 48, 102, 103, 104, 105 และ 106] ตามลำดับ

สำหรับ FIM, MGUG, MPUP, MXUX, MXUPG, RERUN, RN, RG, RP, RPG และ MRPSO ขั้นตอนวิธีเหล่านี้ถูกออกแบบมาให้ประยุกต์ใช้กับปัญหาในรูปแบบของจำนวนจริง ถ้าจะนำขั้นตอนวิธีเหล่านี้มาประยุกต์ใช้กับปัญหาในรูปแบบไบนารีและปัญหาในรูปแบบการสลับลำดับกันในพีชคณิตจะต้องมีขั้นตอนแปลงขั้นตอนวิธี โดยรายละเอียดของขั้นตอนการแปลงขั้นตอนวิธีเหล่านี้ถูกกล่าวไว้ในภาคผนวก ข และ ค ตามลำดับ

ค่าที่ใช้เปรียบเทียบประสิทธิภาพของขั้นตอนวิธีสำหรับการทดลองมีรายละเอียดดังนี้

- ค่าเฉลี่ยของค่าความเหมาะสมที่ดีที่สุด (the mean best fitness value หรือ MBF) [112] คือค่าเฉลี่ยของค่าความเหมาะสมที่ดีที่สุดในรอบสุดท้ายของการค้นหาจากการทดลองทั้งหมด ค่านี้เป็นตัวชี้วัดประสิทธิภาพในการค้นหาคำตอบของขั้นตอนวิธี ในกรณีการทดลองกับฟังก์ชันมาตรฐานจากตาราง ก.1 จะมีจุดต่ำสุดอยู่ที่ศูนย์ทุกฟังก์ชัน เมื่อคำนวณค่าความเหมาะสมได้ค่าที่ต่ำกว่า 10^{-324} จะถูกปิดให้มีค่าเป็นศูนย์ ถ้าหากขั้นตอนวิธีใดที่สามารถหาคำตอบใกล้เคียงกับศูนย์มากที่สุด ขั้นตอนวิธีนั้นก็จะมีประสิทธิภาพในการแก้ปัญหาฟังก์ชันมาตรฐานมากเท่านั้น สำหรับกรณีการทดลองกับปัญหาถุงเป้หลายมิติและปัญหาการเดินทางของพนักงานขายจะมีค่าที่ดีที่สุดซึ่งเป็นที่รู้จักกันโดยทั่วไป (the best known values) ถ้าขั้นตอนวิธีใดที่สามารถหาคำตอบใกล้เคียงกับค่าเหล่านี้ ขั้นตอนวิธีนั้นก็จะมีประสิทธิภาพในการแก้ปัญหาถุงเป้หลายมิติและปัญหาการเดินทางของพนักงานขายมากเท่านั้น
- จำนวนรอบที่ประสบความสำเร็จในการค้นหา (The success round หรือ SR) [112] คือจำนวนรอบจากการทดลองที่การค้นหาสามารถพบคำตอบที่ดีที่สุดของปัญหาที่แก้ไข ค่านี้เป็นตัวชี้วัดความน่าเชื่อถือของขั้นตอนวิธี โดยขั้นตอนวิธีใดที่มีค่านี้นั้นมาก ขั้นตอนวิธีนั้นก็มีความน่าเชื่อถือมากเท่านั้น

- ค่าลอการิทึมเฉลี่ยของค่าความเหมาะสมของอนุภาคทั้งหมด (the logarithm mean fitness values of all particles หรือ LMFx) คือค่าลอการิทึมเฉลี่ยของค่าความเหมาะสมของอนุภาคทั้งหมดในแต่ละรอบของการค้นหา
- ค่าลอการิทึมเฉลี่ยของค่าความเหมาะสมของ PBEST (the logarithm mean fitness values of PBEST หรือ LMFP) คือค่าลอการิทึมเฉลี่ยของค่าความเหมาะสมของ PBEST ทั้งหมดในแต่ละรอบของการค้นหา
- ค่าลอการิทึมเฉลี่ยของค่าความเหมาะสมของ GBEST (the logarithm mean fitness values of GBEST หรือ LMFG) คือค่าลอการิทึมเฉลี่ยของค่าความเหมาะสมของ GBEST ในแต่ละรอบของการค้นหา
- ค่าความเหมาะสมของอนุภาคที่ดีที่สุด (the logarithm fitness values of the Best Particle หรือ LBP) คือค่าลอการิทึมของค่าความเหมาะสมของอนุภาคที่มีค่าดีที่สุดในรอบของการค้นหาปัจจุบัน

โดยค่า LMFx, LMFG, LMFP และ LBP เป็นตัวชี้วัดประสิทธิภาพในการค้นหาของขั้นตอนวิธี ในกรณีการทดลองกับฟังก์ชันมาตรฐานจากตาราง ก.1 จะมีจุดต่ำสุดอยู่ที่ศูนย์ทุกฟังก์ชัน เมื่อคำนวณค่าความเหมาะสมได้ค่าที่ต่ำกว่า 10^{-324} จะถูกปิดให้มีค่าเป็นศูนย์ ถ้าขั้นตอนวิธีใดที่สามารถหาค่าตอบใกล้เคียงกับศูนย์มากกว่าขั้นตอนวิธีนั้นก็จะมีประสิทธิภาพในการแก้ปัญหาฟังก์ชันมาตรฐานได้มากเท่านั้น สำหรับกรณีของปัญหาถุงเป้หลายมิติและปัญหาการเดินทางของพนักงานขายจะมีค่าที่ดีที่สุดซึ่งเป็นที่รู้จักกันโดยทั่วไป ถ้าขั้นตอนวิธีใดที่สามารถหาค่าตอบใกล้เคียงกับค่าเหล่านี้ ขั้นตอนวิธีนั้นก็จะมีประสิทธิภาพในการแก้ปัญหาถุงเป้หลายมิติและปัญหาการเดินทางของพนักงานขายได้มากเท่านั้น

- ค่าเฉลี่ยรอบของการรีเซ็ต (The average restart round หรือ ARR) คือค่าเฉลี่ยของจำนวนการรีเซ็ตจากการทดลองทั้งหมด
- ค่าเฉลี่ยของความเร็วของอนุภาค (The Average Velocity หรือ AVGV) คือค่าเฉลี่ยของความเร็วของอนุภาคจากการทดลองทั้งหมด
- ลอการิทึมของค่าเฉลี่ยของระยะห่างของประชากรระหว่างจุด (The logarithm of population average distance amongst points หรือ LD(t)) คือค่าการกระจายในระดับที่ไม่ต่อเนื่องระหว่างอนุภาคในประชากร ค่านี้เป็นตัวชี้วัดความหลากหลายของประชากร โดยค่านี้ยังมีค่ามากเท่าใด ความหลากหลายของประชากรก็จะยังมีค่ามากเท่านั้น สำหรับ PSO ค่านี้สามารถคำนวณได้จากสมการ 2.4 แต่สำหรับ CM ค่านี้สามารถคำนวณได้จากการผลรวมของสมการ 2.4 และสมการ 5.1

$$D(t) = \frac{1}{M \times L \times RM} \sum_{r=1}^{RM} \sum_{l=1}^M \sqrt{\sum_{d=1}^D (mx'_{ldr} - \overline{mp}_{ldr})^2} \quad (5.1)$$

โดยกำหนดให้	
M	คือ ขนาดของประชากร
RM	คือ รอบของการกลายพันธุ์
L	คือ ความยาวของขอบเขตการค้นหา
D	คือ จำนวนมิติทั้งหมดของปริภูมิคำตอบ
mx'_{idr}	คือ ค่าได้จากการกลายพันธุ์ของมิติ d ของอนุภาค i ในรอบการกลายพันธุ์ที่ r ในรอบการค้นหาคำตอบที่ t
\overline{mp}_{idr}	คือ ค่าเฉลี่ยของมิติที่ได้จากการกลายพันธุ์จากอนุภาคทั้งหมด และรอบการกลายพันธุ์ทั้งหมด
t	คือ รอบการค้นหาคำตอบ

5.1 การทดลองปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO

ในหัวข้อนี้จะพิจารณาถึงความสามารถในการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO เพื่อวิเคราะห์ผลกระทบจากปัญหาการติดในจุดสุดสัมพัทธ์ โดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1 โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหา 10000 รอบ และ 1000 รอบ โดยตาราง 5.4 แสดงค่า MBF และ SR

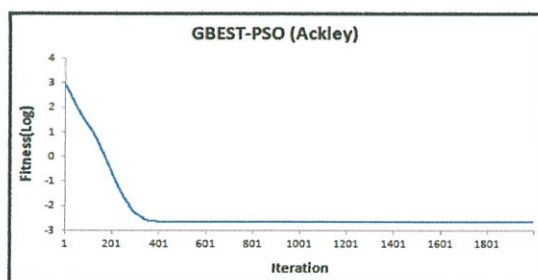
จากตาราง 5.4 สำหรับฟังก์ชันยูนิโมดอล ถ้าเพิ่มรอบการค้นหาจะได้รับคำตอบที่ดีขึ้นอย่างเห็นได้ชัด ซึ่งมีแนวโน้มว่าคำตอบจะดีขึ้นเรื่อยๆ แล้วมีโอกาสสูงที่จะพบคำตอบที่ดีที่สุด ดังนั้น PSO จึงมีประสิทธิภาพที่ดีสำหรับการแก้ปัญหายูนิโมดอล เพราะสามารถหาคำตอบที่ดีที่สุดพบ แต่สำหรับปัญหามัลติโมดอลการเพิ่มรอบการค้นหาโดยส่วนมากแล้วไม่สามารถได้รับคำตอบที่ดีขึ้น เพราะ PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ง่าย PSO จึงมีประสิทธิภาพต่ำสำหรับการแก้ปัญหา มัลติโมดอล

เพื่อแสดงให้เห็นผลกระทบจากปัญหาการติดในจุดสุดสัมพัทธ์ จึงได้ทำการทดลองโดยการค้นหาคำตอบด้วย PSO กับฟังก์ชัน ACKLEY ซึ่งเป็นตัวแทนปัญหามัลติโมดอลและฟังก์ชัน SPHERE ซึ่งเป็นตัวแทนของปัญหายูนิโมดอล โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.1 ถึง 5.6 ใช้จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟ 5.1 และ 5.2 ได้แสดงค่า LMFG ของการค้นหาด้วย PSO ของฟังก์ชัน ACKLEY และฟังก์ชัน SPHERE ตามลำดับ ส่วนกราฟ 5.3 และ 5.4 แสดงค่า AVGV ของการค้นหาด้วย PSO ของฟังก์ชัน ACKLEY และฟังก์ชัน SPHERE ตามลำดับ สำหรับกราฟ 5.5 และ 5.6 ได้แสดงค่า LD(t) ของการค้นหาด้วย PSO ของฟังก์ชัน ACKLEY และฟังก์ชัน SPHERE ตามลำดับ

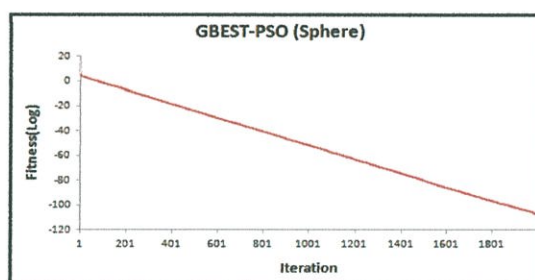
จากผลการทดลองสามารถสรุปได้ว่า PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ง่าย แต่ผลจากการที่ PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ง่ายทำให้สามารถตรวจสอบการเกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ง่ายเช่นกัน โดยเมื่อ PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ GBEST จะมีค่าคงที่เป็นระยะเวลานานดังรูป 5.1

ตารางที่ 5.4 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO

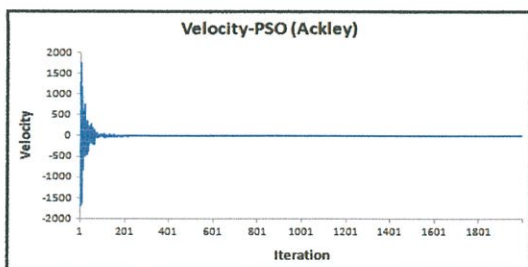
ALGORITHM	PSO (1000)		PSO(10000)	
	MBF	SR	MBF	SR
ACKLEY	0.0727646	0	0.0399113	0
GRIEWANK	0.00844392	40	0.00774393	47
RASTRIGIN	99.6846	0	107.475	0
ROSENBROCK	1.43E-15	0	4.56E-31	97
SCHWEFEL	2431.05	0	2467.96	0
COSINE	0.37829	4	0.393068	8
EXPONENTIAL	1.81E-16	0	1.41E-16	0
LEVY	7.5432	0	7.08389	2
MICHALEWICZ	0.525012	0	0.442718	0
STEP	54.59	0	53.26	0
SCHAFFER	0.000194318	98	0	100
HOLDER	0	100	0	100
BEALE	0	100	0	100
SHUBERT	0	100	0	100
GOLDSTEIN	0	100	0	100
TRID	4.34E-12	12	3.18E-12	17
SPHERE	1.75E-22	0	3.00E-237	0
PARALLEL	1.58E-21	0	6.64E-237	0
ROTATED	1.50E-19	0	3.26E-239	0
CIGAR	5.96E-17	0	2.72E-232	0
BROWN	3.32E-23	0	7.36E-238	0
MULTIMOD	42.8241	0	4.27725	0
ZAKHAROV	0.145695	0	2.46E-35	0
EASOM	0	100	0	100
ผลรวม		654		771



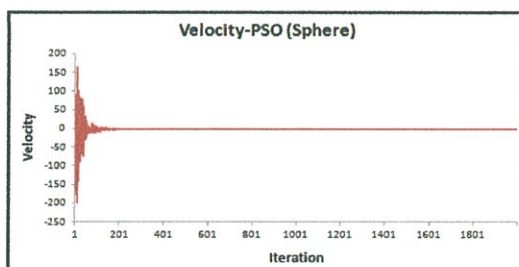
รูปที่ 5.1 LMFG ของฟังก์ชัน ACKLEY ที่ค้นหา
คำตอบด้วย PSO



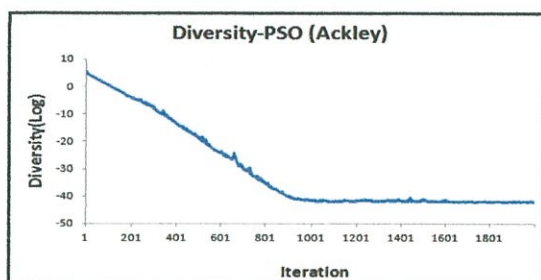
รูปที่ 5.2 LMFG ของฟังก์ชัน SPHERE ที่ค้นหา
คำตอบด้วย PSO



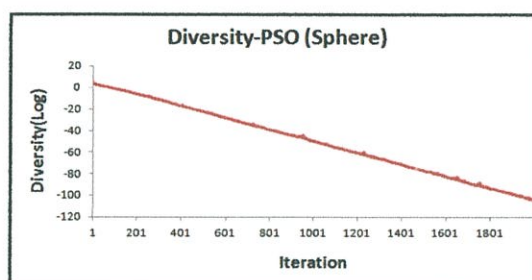
รูปที่ 5.3 AVGV ของฟังก์ชัน ACKLEY ที่ค้นหา
คำตอบด้วย PSO



รูปที่ 5.4 AVGV ของฟังก์ชัน SPHERE ที่ค้นหา
คำตอบด้วย PSO



รูปที่ 5.5 LD(t) ของฟังก์ชัน ACKLEY ที่ค้นหา
คำตอบด้วย PSO



รูปที่ 5.6 LD(t) ของฟังก์ชัน SPHERE ที่ค้นหา
คำตอบด้วย PSO

จากกราฟ 5.1 หลังจากการค้นหาผ่านไประยะเวลาหนึ่ง คำตอบที่ได้จะไม่ดีขึ้นเลยและไม่ว่าเวลาจะผ่านไปนานเท่าใด คำตอบที่ได้ก็ไม่มีแนวโน้มว่าจะดีขึ้น ซึ่งเหตุการณ์นี้เรียกว่าปัญหาการติดในจุดสุดสัมพัทธ์ แต่กราฟ 5.2 คำตอบที่ได้มีค่าดีขึ้นเรื่อยๆ ซึ่งมีแนวโน้มว่าจะพบคำตอบที่ดีที่สุด

จากกราฟ 5.3 และ 5.4 แสดงให้เห็นว่าอนุภาคจะลดการค้นหาในบริเวณกว้างมาเป็นบริเวณแคบแทน เพราะความเร็วของอนุภาคลดลงจนเกือบเป็นศูนย์ อนุภาคจึงแทบจะไม่มี การเคลื่อนที่เลย ดังนั้นคำตอบที่ค้นหาจึงอยู่ในบริเวณแคบและไม่มีโอกาสที่จะไปค้นหาบริเวณอื่น การเคลื่อนที่แบบนี้ จะช่วยให้ลู่เข้าไปหาคำตอบสำหรับปัญหาชนิดโมดอล แต่ปัญหามัลติโมดอลผลจากความเร็วลดลงนี้ จะทำให้ลู่เข้าไปค้นหาในบริเวณจุดสุดสัมพัทธ์แล้วไม่ออกไปค้นหาในบริเวณอื่นอีกจึงเกิดปัญหาการติดในจุดสุดสัมพัทธ์

จากกราฟ 5.5 แสดงให้เห็นว่าความหลากหลายของอนุภาคลดลงและจะคงที่เมื่ออนุภาคติดในจุดสุดสัมพัทธ์ สำหรับกรณีปัญหามัลติโมดอลผลจากความหลากหลายของอนุภาคลดลง อนุภาคค้นหาในบริเวณจุดสุดสัมพัทธ์แล้วไม่ไปค้นหาในบริเวณอื่นอีกจึงเกิดปัญหาการติดในจุดสุดสัมพัทธ์ สำหรับกราฟ 5.6 แสดงให้เห็นว่าความหลากหลายของอนุภาคลดลงเรื่อยๆ อนุภาคจึงค้นหาในบริเวณที่แคบลงเรื่อยๆ ซึ่งทำให้อนุภาคลู่เข้าหาคำตอบสำหรับกรณีปัญหาชนิดโมดอล

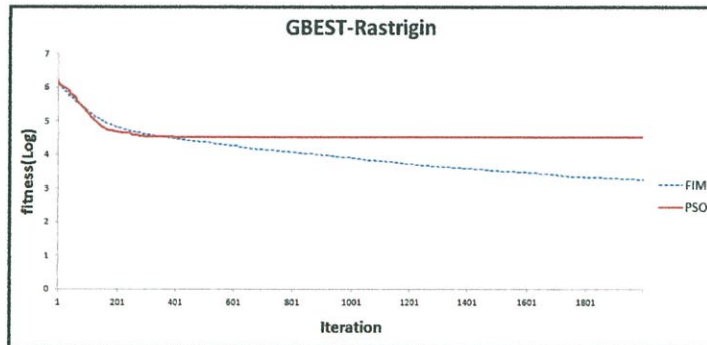
5.2 การทดลองการประยุกต์ใช้การกลายพันธุ์กับ PSO

ในหัวข้อนี้พิจารณาถึงความสามารถในการค้นหาคำตอบฟังก์ชันมาตรฐานด้วย PSO, MXUX ซึ่งเป็นตัวแทนของการกลายพันธุ์แบบ CM และ FIM ซึ่งเป็นตัวแทนของการกลายพันธุ์แบบ NCM เพื่อพิจารณาประสิทธิภาพของการประยุกต์ใช้การกลายพันธุ์กับ PSO และการวิเคราะห์ผลกระทบจากการกลายพันธุ์กับ PSO โดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1

5.2.1 การพิจารณาการกลายพันธุ์แบบ NCM กับ PSO

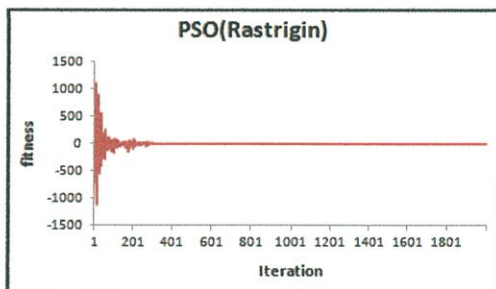
หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย PSO และ FIM กับฟังก์ชัน RASTRIGIN และ LEVY ซึ่งเป็นตัวแทนของปัญหามัลติโมดอล โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.7 ถึงกราฟ 5.10 ได้ใช้จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟ 5.7 ได้แสดงค่า LMFG ของการค้นหาด้วย PSO และ FIM ของฟังก์ชัน RASTRIGIN ส่วนกราฟ 5.8 และ 5.9 แสดงค่า AVGV ของการค้นหาด้วย PSO และ FIM ของฟังก์ชัน RASTRIGIN ตามลำดับ สำหรับกราฟ 5.10 ได้แสดง

ค่า $LD(t)$ ของการค้นหาคำด้วย PSO และ FIM ของฟังก์ชัน LEVY และสำหรับตาราง 5.5 ได้ใช้พารามิเตอร์ตามตาราง 5.1 โดยตารางแสดงค่า MBF และ SR

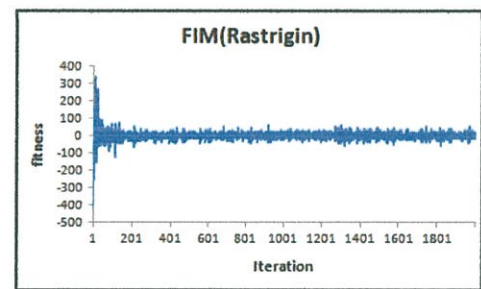


รูปที่ 5.7 LMFG ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย PSO และ FIM

กราฟ 5.7 เส้นประแทน FIM และเส้นทึบแทน PSO แสดงให้เห็นว่า PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์โดยคำตอบที่ได้จะไม่ดีขึ้นเลยซึ่งจะเห็นได้ว่าเส้นกราฟมีค่าคงที่ แต่ FIM จะไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยคำตอบจะดีขึ้นเรื่อยๆ ซึ่งจะเห็นได้ว่าเส้นกราฟมีค่าลดลงไปเรื่อยๆ จึงมีแนวโน้มที่จะพบคำตอบที่ดีที่สุด จากผลการทดลองสรุปว่าการกลายพันธุ์สามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้

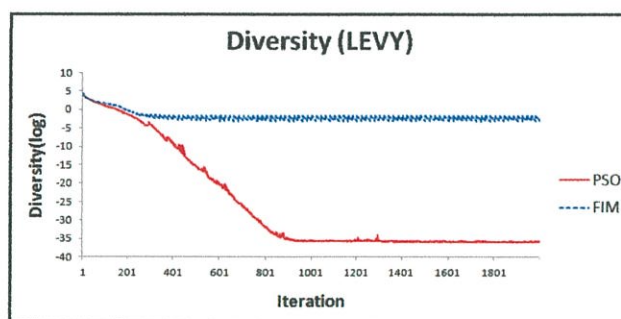


รูปที่ 5.8 AVGV ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย PSO



รูปที่ 5.9 AVGV ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย FIM

จากกราฟ 5.8 จะเห็นว่าความเร็วของอนุภาคลดลงจนเกือบเป็นศูนย์ ซึ่งอนุภาคจะเคลื่อนที่ในบริเวณแคบๆ แต่กราฟ 5.9 จะเห็นว่าความเร็วของอนุภาคเกิดเปลี่ยนแปลงตลอดเวลาเนื่องจากผลของการกลายพันธุ์ทำให้อนุภาคกระโดดไปยังบริเวณอื่นๆ จากผลการทดลองสรุปว่าการกลายพันธุ์เพิ่มการกระจายตัวให้กับอนุภาคทำให้สามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้



รูปที่ 5.10 $D(t)$ ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย PSO และ FIM

จากกราฟ 5.10 จะเห็นว่าเส้นกราฟของ PSO ในช่วงคงที่มีค่าน้อยกว่าเส้นกราฟของ FIM ในช่วงคงที่ หมายความว่าในขณะที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ความหลากหลายประชากรของ PSO จะมีค่าน้อยกว่าความหลากหลายประชากรของ FIM ส่งผลให้ FIM มีขอบเขตการค้นหาที่กว้างกว่า PSO จึงมีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ได้มากกว่า

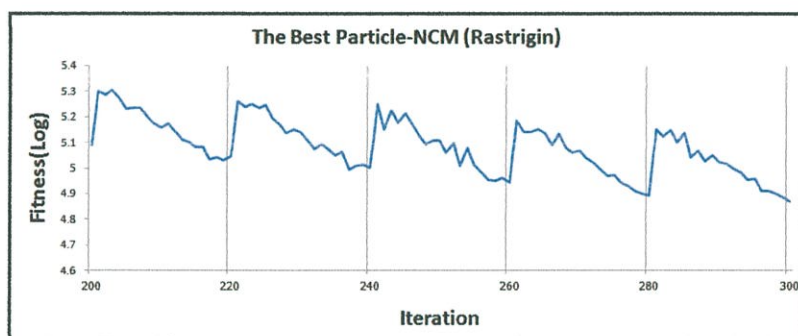
ตารางที่ 5.5 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO และ FIM

ALGORITHM	PSO		FIM	
	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	7.53E-15	0
GRIEWANK	0.00736214	48	0.000542335	93
RASTRIGIN	104.351	0	1.53E-14	78
ROSENBROCK	3.94E-31	98	0.0553273	0
SCHWEFEL	2382.29	0	1867.79	0
COSINE	0.385717	8	0	100
EXPONENTIAL	1.25E-16	0	1.11E-16	0
LEVY	6.51724	3	2.90E-05	0
MICHALEWICZ	0.417773	0	0.071792	92
STEP	39.04	0	0	100
SCHAFFER	0	100	0	100
HOLDER	0	100	0	100
BEALE	0	100	0	100
SHUBERT	0	100	0	100
GOLDSTEIN	0	100	0	100
TRID	3.29E-12	19	4.03E-10	0
SPHERE	0	100	0	100
PARALLEL	0	100	0	100
ROTATED	0	100	0	100
CIGAR	0	100	0	100
BROWN	0	100	0	100
MULTIMOD	0	100	0	100
ZAKHAROV	3.35E-188	0	2.10E-199	0
EASOM	0	100	0	100
ผลรวม		1376		1663

จากตาราง 5.5 ผลการทดลองของปัญหาหมัลติโมดอลพบว่า FIM ได้ผลลัพธ์โดยรวมดีกว่า PSO เพราะ FIM สามารถเพิ่มความหลากหลายให้กับประชากร จึงสามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า PSO ดังนั้นสำหรับปัญหาหมัลติโมดอลการกลายพันธุ์แบบ NCM สามารถช่วยปรับปรุงคำตอบและเพิ่มประสิทธิภาพการค้นหาของ PSO แต่สำหรับผลการทดลองของปัญหายูนิโมดอลพบว่า FIM ได้ผลลัพธ์โดยรวม แย่กว่า PSO เพราะการกลายพันธุ์ทำให้อุณหภูมิกระจายตัวจึงขัดขวาง การลู่เข้าหากันของอนุภาค ดังนั้นการกลายพันธุ์แบบ NCM อาจจะสามารถทำให้ PSO มีประสิทธิภาพการค้นหาลดลงสำหรับปัญหายูนิโมดอล

5.2.2 การพิจารณาการเพิ่มประสิทธิภาพการค้นหาของ PSO ด้วย NCM

หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย FIM กับฟังก์ชัน RASTRIGIN โดยใช้พารามิเตอร์ตามตารางที่ 5.1 แต่สำหรับกราฟ 5.11 ได้ใช้จำนวนรอบการค้นหาคำตอบ 2000 รอบ จำนวนครั้งในการทดลอง 1 ครั้งเท่านั้น ได้แสดงค่าของ LBP ของการค้นหาด้วย FIM ของฟังก์ชัน RASTRIGIN ซึ่งถูกเลือกเป็นตัวแทนของปัญหาหมัลติโมดอลในรอบการค้นหาช่วง 200 ถึง 300 รอบ



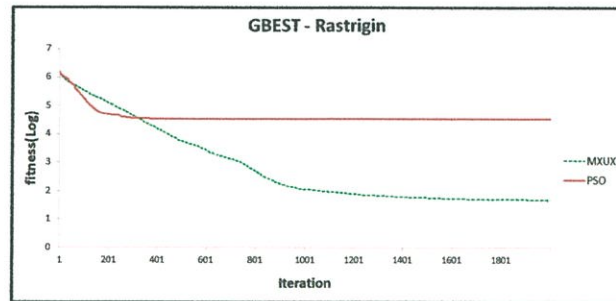
รูปที่ 5.11 FIM ปรับปรุง PSO ในช่วงรอบการค้นหา 200 ถึง 300

จากรูป 5.11 การกลายพันธุ์ถูกกำหนดให้ดำเนินการทุก 20 รอบของการค้นหา โดยการกลายพันธุ์ทำให้เกิดการกระจายตัวของอนุภาคออกจากตำแหน่งของ GBEST ไปยังตำแหน่งที่แย่กว่าเดิมสังเกตได้จากเส้นกราฟเมื่อผ่านการกลายพันธุ์เส้นกราฟจะมีค่าเพิ่มขึ้น หลังจากนั้นอนุภาคจะทำการลู่เข้าหากันใหม่ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้คำตอบนั้นดีขึ้นสังเกตได้จากเส้นกราฟที่จะลู่ลงไปมากกว่าก่อนการกลายพันธุ์ จากผลการทดลองสรุปว่าการกลายพันธุ์ทำให้อุณหภูมิกระจายตัวออกจากกันแล้วการปรับปรุงคำตอบเกิดจากการที่อนุภาคลู่เข้าหากัน

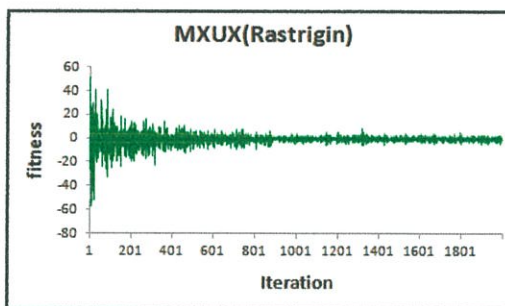
5.2.3 การพิจารณาการกลายพันธุ์แบบ CM

หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย PSO และ MXUX กับฟังก์ชัน RASTRIGIN และ LEVY ซึ่งเป็นตัวแทนของปัญหาหมัลติโมดอลโดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.12 ถึง 5.14 ได้ใช้จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟ 5.12 ได้แสดงค่า LMFG ของการค้นหาด้วย PSO และ MXUX ของฟังก์ชัน RASTRIGIN ส่วนกราฟ 5.13 แสดงค่า AVGV ของการ

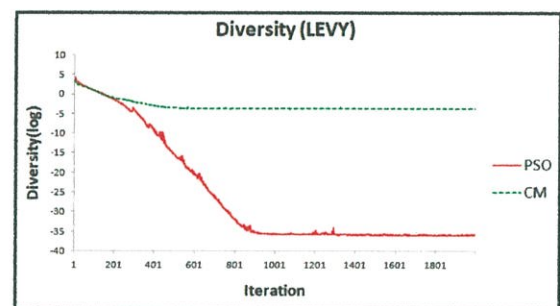
ค้นหาด้วย MXUX ของฟังก์ชัน RASTRIGIN ส่วนกราฟ 5.14 ได้แสดงค่าลอการิทึมของ $D(t)$ ของการค้นหาด้วย PSO และ FIM ของฟังก์ชัน LEVY



รูปที่ 5.12 LMFG ของฟังก์ชัน RASTRIGIN ที่ค้นหาด้วย PSO และ MXUX



รูปที่ 5.13 AVGV ของฟังก์ชัน RASTRIGIN ที่ค้นหาด้วย MXUX



รูปที่ 5.14 $D(t)$ ของฟังก์ชัน LEVY ที่ค้นหาด้วย PSO และ MXUX

จากกราฟ 5.12 เส้นประแทน MXUX และเส้นทึบแทน PSO จากกราฟจะเห็นว่า PSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยคำตอบที่ได้จะไม่ดีขึ้นเลยซึ่งจะเห็นได้ว่าเส้นกราฟมีค่าคงที่ แต่ MXUX จะไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์โดยคำตอบจะดีขึ้นเรื่อยๆ ซึ่งจะเห็นได้ว่าเส้นกราฟมีค่าลดลงจึงมีแนวโน้มที่จะพบคำตอบที่ดีที่สุด จากผลการทดลองสรุปว่าการกลายพันธุ์สามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้

จากกราฟ 5.13 จะเห็นว่าความเร็วของอนุภาคเกิดเปลี่ยนแปลงตลอดเวลาเนื่องจากผลของการกลายพันธุ์ ทำให้อนุภาคกระโดดไปยังบริเวณอื่นๆ จากผลการทดลองสรุปว่าการกลายพันธุ์เพิ่มการกระจายตัวให้กับอนุภาค ทำให้สามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้

จากกราฟ 5.14 จะเห็นว่าเส้นกราฟของ PSO ในช่วงคั้งที่นั้นมีค่าน้อยกว่าเส้นกราฟของ MXUX ในช่วงคั้งที่หมายความว่าในขณะที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ความหลากหลายประชากรของ PSO จะมีค่าน้อยกว่าความหลากหลายประชากรของ MXUX ส่งผลให้ MXUX มีขอบเขตการค้นหาที่กว้างกว่า PSO จึงมีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ได้มากกว่า

สำหรับตาราง 5.6 ได้ใช้พารามิเตอร์ตามตาราง 5.1 โดยตารางแสดงค่า MBF และ SR จากตารางนี้เมื่อพิจารณาผลการทดลองของปัญหาโมดอลพบว่า MXUX ได้ผลลัพธ์โดยรวมดีกว่า PSO เพราะ MXUX สามารถเพิ่มความหลากหลายให้กับประชากร จึงสามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า PSO ดังนั้นการกลายพันธุ์แบบ CM สามารถช่วยปรับปรุงคำตอบและเพิ่มประสิทธิภาพการค้นหาของ PSO สำหรับปัญหาโมดอลบางปัญหา PSO ได้

ผลลัพธ์ดีกว่า MXUX เช่น ROSENBROCK, SCHWEFEL และ TRID ดังนั้นในบางปัญหา MXUX อาจจะทำให้ผลลัพธ์การค้นหาที่แย่กว่า PSO

ตารางที่ 5.6 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO และ MXUX

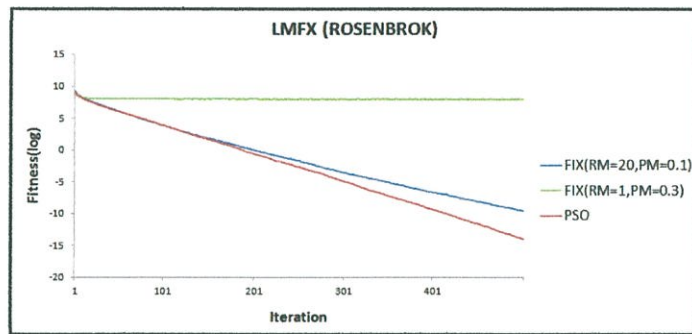
ALGORITHM	PSO		MXUX	
	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	7.11E-15	0
GRIEWANK	0.00736214	48	0	100
RASTRIGIN	104.351	0	0	100
ROSENBROCK	3.94E-31	98	1.31E-25	0
SCHWEFEL	2382.29	0	2496.14	0
COSINE	0.385717	8	0	100
EXPONENTIAL	1.25E-16	0	1.10E-16	1
LEVY	6.51724	3	0.312454	5
MICHALEWICZ	0.417773	0	0.238974	78
STEP	39.04	0	0	100
SCHAFFER	0	100	0	100
HOLDER	0	100	0	100
BEALE	0	100	0.0152414	98
SHUBERT	0	100	0	100
GOLDSTEIN	0	100	0	100
TRID	3.29E-12	19	1.37E-11	1
SPHERE	0	100	0	100
PARALLEL	0	100	0	100
ROTATED	0	100	0	100
CIGAR	0	100	0	100
BROWN	0	100	0	100
MULTIMOD	0	100	0	100
ZAKHAROV	3.35E-188	0	0	100
EASOM	0	100	0	100
ผลรวม		1376		1783

5.3 การทดลองปัญหาจากการกลายพันธุ์แบบ NCM

ในหัวข้อนี้จะพิจารณาถึงความสามารถในการค้นหาคำตอบจากฟังก์ชันมาตรฐานด้วย PSO และ FIM เพื่อพิจารณาผลกระทบจากปัญหาจากการกลายพันธุ์แบบ NCM โดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1

5.3.1 การพิจารณาปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย FIM กับฟังก์ชัน ROSENBROCK โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟที่ 5.15 จำนวนรอบการค้นหาคำตอบ 500 รอบ รอบของการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 1 รอบ และ 20 รอบ ความน่าจะเป็นในการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 0.30 และ 0.10 โดยกราฟแสดงค่า LMFx ของการค้นหาด้วย FIM และ PSO ของฟังก์ชัน ROSENBROCK ซึ่งถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอล สำหรับตาราง 5.7 รอบของการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 1 รอบ ความน่าจะเป็นในการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 0.30 โดยตารางได้ใช้พารามิเตอร์ตามตาราง 5.1 และได้แสดงค่า MBF และ SR



รูปที่ 5.15 LMFx ของฟังก์ชัน ROSENBRACK ที่ค้นหาคำตอบด้วย PSO

จากกราฟ 5.15 เส้นสีเขียวแทน FIM มีรอบของการกลายพันธุ์เท่ากับ 1 รอบ มีความน่าจะเป็นในการกลายพันธุ์เท่ากับ 0.30 เส้นสีฟ้าแทน FIM มีรอบของการกลายพันธุ์เท่ากับ 20 รอบ มีความน่าจะเป็นในการกลายพันธุ์เท่ากับ 0.10 เส้นสีแดงแทน PSO โดยจากกราฟจะเห็นว่า ถ้าดำเนินการกลายพันธุ์มากเกินไป (1 รอบ และ 30 เปอร์เซ็นต์) จะส่งผลให้อนุภาคไม่ลู่เข้าหากัน คำตอบที่ได้จึงแย่กว่าไม่ดำเนินการกลายพันธุ์ เพราะเส้นกราฟจะมีแนวโน้มคงที่เมื่อเปรียบกับการดำเนินการกลายพันธุ์ที่เหมาะสม (20 รอบ และ 10 เปอร์เซ็นต์) ซึ่งได้ผลลัพธ์ดีกว่า เพราะเส้นกราฟนั้นลู่ลงไปเรื่อยๆ เช่นเดียวกับ PSO ที่เส้นกราฟสามารถลู่ลงไปเรื่อยๆ จากกราฟแสดงให้เห็นว่าการกลายพันธุ์มากเกินไปจะขัดขวางการลู่เข้าหากันของอนุภาค ทำให้ผลลัพธ์การค้นหาแย่กว่า PSO หรือการดำเนินการกลายพันธุ์ที่เหมาะสม

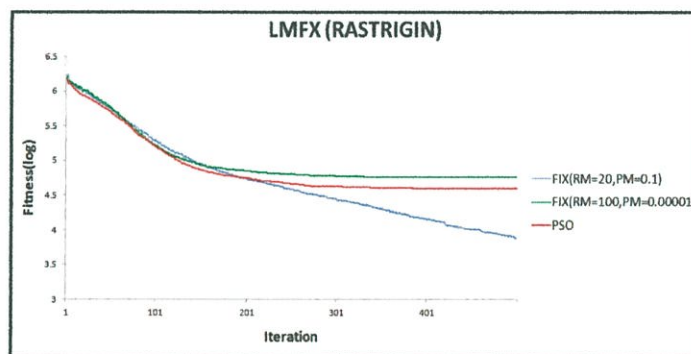
ตารางที่ 5.7 ผลการทดลองปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

ALGORITHM	PSO		FIM (RM = 1, PM = 0.30)	
	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	3.94E-15	0
GRIEWANK	0.00736214	48	0	100
RASTRIGIN	104.351	0	0	100
ROSENBRACK	3.94E-31	98	1418.4	0
SCHWEFEL	2382.29	0	5712.58	0
COSINE	0.385717	8	0	100
EXPONENTIAL	1.25E-16	0	1.11E-18	99
LEVY	6.51724	3	2.13307	0
MICHALEWICZ	0.417773	0	2.19624	0
STEP	39.04	0	236.18	0
SCHAFFER	0	100	0	100
HOLDER	0	100	3.18E-05	7
BEALE	0	100	9.80E-07	0
SHUBERT	0	100	0.00568356	0
GOLDSTEIN	0	100	4.90E-08	0
TRID	3.29E-12	19	21.2871	0
SPHERE	0	100	1.95E-219	0
PARALLEL	0	100	1.15E-219	0
ROTATED	0	100	4.19E-227	0
CIGAR	0	100	5.54E-213	0
BROWN	0	100	2.90E-220	0
MULTIMOD	0	100	0	100
ZAKHAROV	3.35E-188	0	3.66E-37	0
EASOM	0	100	2.30E-05	0
ผลรวม		1376		606

จากตารางผลการทดลอง 5.7 แสดงให้เห็นว่าสำหรับปัญหาอัลติโมดอลการกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป (1 รอบ และ 30 เปอร์เซ็นต์) ผลลัพธ์จากการค้นหาโดยรวมแล้วแย่กว่า PSO และสำหรับปัญหายูนิโมดอลการกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป ผลลัพธ์ทั้งหมดแย่กว่า PSO เนื่องจากการกลายพันธุ์มากเกินไปจะขัดขวางการลู่เข้าหากันของอนุภาคทำให้ผลลัพธ์การค้นหาแย่กว่า PSO

5.3.2 การพิจารณาปัญหาการกลายพันธุ์น้อยเกินไปของ NCM

หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย FIM กับฟังก์ชัน RASTRIGIN โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.16 จำนวนรอบการค้นหาคำตอบ 500 รอบ รอบของการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 20 รอบ และ 100 รอบ ความน่าจะเป็นในการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 0.10 และ 0.00001 โดยกราฟแสดงค่า LMFx ของการค้นหาด้วย FIM และ PSO ของฟังก์ชัน RASTRIGIN ซึ่งถูกเลือกเป็นตัวแทนของปัญหาอัลติโมดอลสำหรับตาราง 5.8 รอบของการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 2000 รอบ ความน่าจะเป็นในการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 0.00001 โดยตารางได้ใช้พารามิเตอร์ตามตาราง 5.1 และได้แสดงค่า MBF และ SR



รูปที่ 5.16 LMFx ของฟังก์ชัน RASTRIGIN ที่ค้นหาคำตอบด้วย FIM และ PSO

จากกราฟ 5.16 เส้นสีเขียวแทน FIM มีรอบของการกลายพันธุ์ของเท่ากับ 100 รอบ มีความน่าจะเป็นในการกลายพันธุ์เท่ากับ 0.00001 เส้นสีฟ้าแทน FIM มีรอบของการกลายพันธุ์ของเท่ากับ 20 รอบ มีความน่าจะเป็นในการกลายพันธุ์เท่ากับ 0.10 เส้นสีแดงแทน PSO จากกราฟการดำเนินการกลายพันธุ์น้อยเกินไป (100 รอบ และ 0.00001 เปอร์เซ็นต์) ทำให้การค้นหายังคงประสบปัญหาการติดในจุดสุดสัมพัทธ์เช่นเดียวกับ PSO สังเกตจากเส้นกราฟจากการกลายพันธุ์ที่น้อยเกินไป และ PSO ในช่วงแรกของการค้นหาเส้นกราฟจะมีค่าลดลง แต่เมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ เส้นกราฟทั้งสองก็เข้าสู่สภาพคงที่ ในขณะที่การกลายพันธุ์ที่เหมาะสม (20 รอบ และ 0.1 เปอร์เซ็นต์) สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้ โดยสังเกตจากเส้นกราฟที่มีค่าลดลงเรื่อยๆ ดังนั้นการกลายพันธุ์น้อยเกินไปไม่สามารถช่วยแก้ปัญหาการติดในจุดสุดสัมพัทธ์

จากตารางผลการทดลอง 5.8 แสดงให้เห็นว่าถ้ากำหนดให้ดำเนินการกลายพันธุ์น้อยเกินไป (2000 รอบ และ 0.00001 เปอร์เซ็นต์) ผลลัพธ์ที่ได้จากการค้นหานั้นจะไม่ค่อยแตกต่างจากการค้นหาของ PSO เนื่องจากการกลายพันธุ์ที่เพิ่มเข้าไปไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์

ตารางที่ 5.8 ผลการทดลองปัญหาการกลายพันธุ์น้อยเกินไปสำหรับ NCM

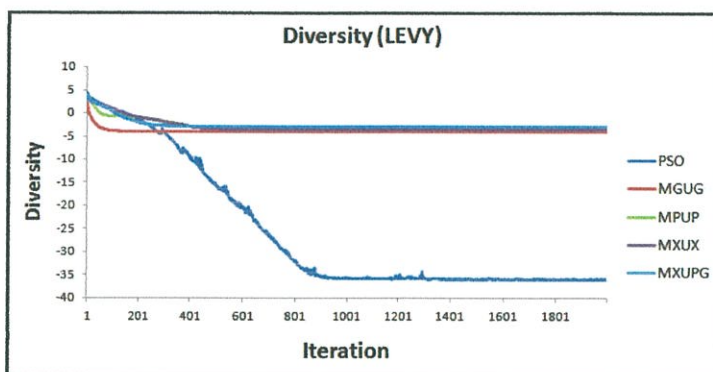
ALGORITHM	PSO		FIM (RM = 2000, PM = 0.00001)	
	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	0.0635115	0
GRIEWANK	0.00736214	48	0.00800028	42
RASTRIGIN	104.351	0	107.117	0
ROSENBROCK	3.94E-31	98	1.41E-30	93
SCHWEFEL	2382.29	0	2420.58	0
COSINE	0.385717	8	0.360518	9
EXPONENTIAL	1.25E-16	0	1.22E-16	0
LEVY	6.51724	3	7.18114	1
MICHALEWICZ	0.417773	0	0.429138	0
STEP	39.04	0	45.89	0
SCHAFFER	0	100	0	100
HOLDER	0	100	0	100
BEALE	0	100	0	100
SHUBERT	0	100	0	100
GOLDSTEIN	0	100	0	100
TRID	3.29E-12	19	2.88E-12	17
SPHERE	0	100	0	100
PARALLEL	0	100	0	100
ROTATED	0	100	0	100
CIGAR	0	100	0	100
BROWN	0	100	0	100
MULTIMOD	0	100	0	100
ZAKHAROV	3.35E-188	0	7.95E-188	0
EASOM	0	100	0	100
ผลรวม		1376		1362

5.4 การทดลองปัญหาและแนวทางการแก้ปัญหาการกลายพันธุ์แบบ CM

ในหัวข้อนี้จะพิจารณาถึงความสามารถในการค้นหาคำตอบจากฟังก์ชันมาตรฐานด้วย PSO, MXUX, MPUP, MGUG และ MXUPG เพื่อวิเคราะห์ผลกระทบจากปัญหาการกลายพันธุ์แบบ CM และแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM โดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1

5.4.1 การพิจารณาความหลากหลายของประชากรของการกลายพันธุ์แบบ CM

หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย PSO, MXUX, MPUP, MGUG และ MXUPG กับฟังก์ชัน LEVY โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.17 จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟแสดงค่า LD(t) ของการค้นหาด้วย PSO, MXUX, MPUP, MGUG และ MXUPG ของฟังก์ชัน LEVY ซึ่งถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอล

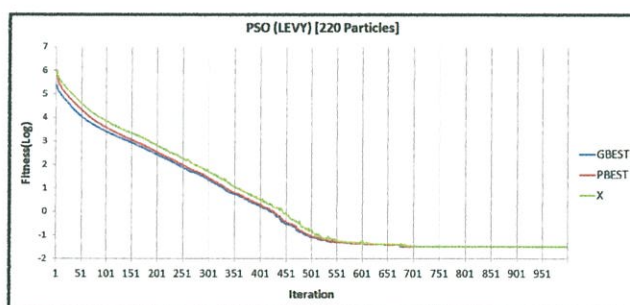


รูปที่ 5.17 ความหลากหลายของประชากรของ PSO และ CM ของฟังก์ชัน LEVY

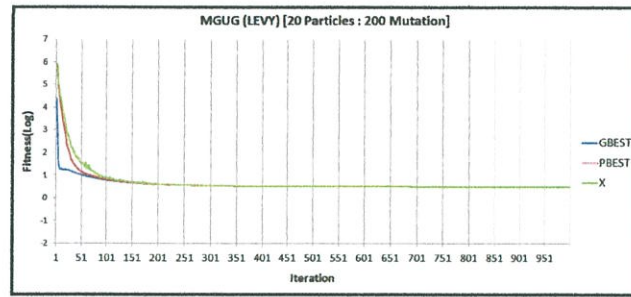
จากกราฟ 5.17 แสดงให้เห็นว่าความหลากหลายของประชากรในช่วงเริ่มต้นของการค้นหาของ PSO, MGUG, MPUP, MXUX และ MXUPG จะมีค่ามาก เมื่อเวลาผ่านไปความหลากหลายของประชากรจะมีค่าลดลง เพราะอนุภาคพยายามลู่เข้าหากันตามหลักการทำงานของ PSO และเมื่ออนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ความหลากหลายของประชากรจะอยู่ในระดับคงที่ ในช่วงที่ระดับคงที่เส้นกราฟของ MGUG, MPUP, MXUX และ MXUPG จะมีระดับใกล้เคียงกันและมีระดับที่มากกว่า PSO หมายความว่าในช่วงที่อนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ ความหลากหลายของประชากรของ MGUG, MPUP, MXUX และ MXUPG จะมีค่าใกล้เคียงกันและมีค่ามากกว่าความหลากหลายของประชากรของ PSO

5.4.2 การพิจารณาปัญหาการลู่เร็วเกินไปและแนวทางแก้ปัญหาการลู่เร็วเกินไป

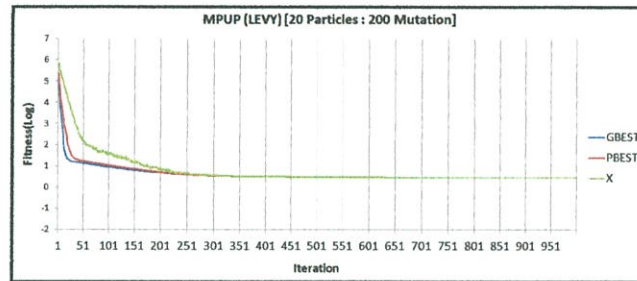
หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย PSO, MGUG, MPUP, MXUX และ MXUPG กับฟังก์ชัน LEVY โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.18 ถึง 5.22 จำนวนรอบการค้นหาคำตอบ 1000 รอบ โดยกราฟแสดงค่า LMFx, LMFP และ LMFG ของการค้นหาด้วย PSO, MGUG, MPUP, MXUX และ MXUPG ของฟังก์ชัน LEVY ซึ่งถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอล



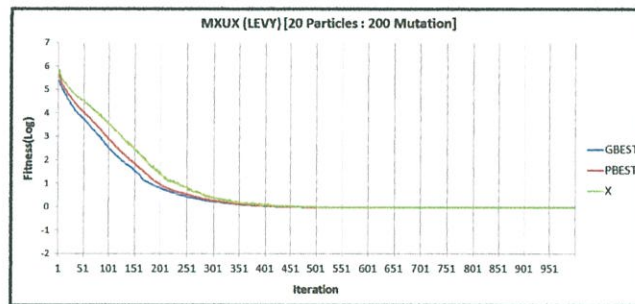
รูปที่ 5.18 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย PSO



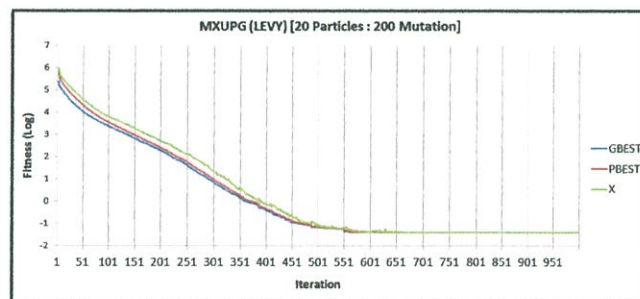
รูปที่ 5.19 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MGUG



รูปที่ 5.20 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MPUP



รูปที่ 5.21 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MXUX



รูปที่ 5.22 LMFx, LMFP และ LMFG ของฟังก์ชัน LEVY ที่ค้นหาคำตอบด้วย MXUPG

จากกราฟ 5.18 ถึง 5.22 ในช่วงแรกของการค้นหาเส้นกราฟของ LMFx, LMFP, LMFG ของ PSO, MGUG, MPUP, MXUX, MXUPG จะมีค่ามากแล้วจะลดลงเรื่อยๆ เพราะอนุภาคพยายามลู่ออกกันตามหลักการทำงานของ PSO และเมื่ออนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ เส้นกราฟทั้ง 3 จะเข้าสู่สถานะคงที่จนสิ้นสุดการค้นหาเส้นกราฟของ LMFx ของ MGUG, MPUP, MXUX รอบของการค้นหาเข้าสู่สถานะคงที่ประมาณ 150, 300 และ 450 รอบตามลำดับ ในขณะที่เส้นกราฟของ LMFx ของ PSO และ MXUPG รอบของการค้นหาเข้าสู่สถานะคงที่ประมาณ 700

รอบ และ 650 รอบตามลำดับ ซึ่ง MGUG, MPUP, MXUX เข้าสู่สภาพคงที่เร็วกว่า PSO และ MXUPG หมายความว่า MGUG, MPUP, MXUX มีความเร็วในการลู่เข้าหากันเร็วกว่า PSO และ MXUPG ซึ่งทำให้ MGUG, MPUP, MXUX เกิดปัญหาการลู่เร็วเกินไปซึ่งจะทำให้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ได้ง่ายกว่า PSO และ MXUPG ดังนั้นการกลายพันธุ์ช่วยเร่งให้อนุภาคลู่เข้าหากันเร็วกว่า PSO ซึ่งเป็นเหตุให้เกิดปัญหาการลู่เร็วเกินไปมากกว่า PSO แต่ MXUPG มีความเร็วในการลู่เข้าหากันใกล้เคียงกับ PSO จึงแทบจะไม่เพิ่มปัญหาการลู่เร็วเกินไป

ตารางที่ 5.9 ผลการทดลองปัญหาการลู่เร็วเกินไป

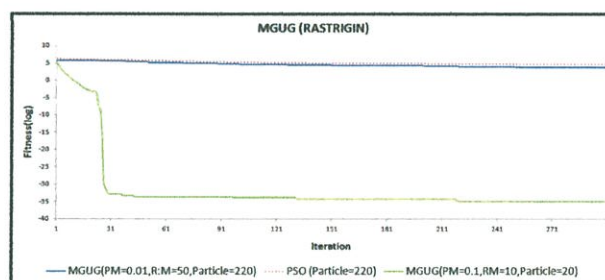
ALGORITHM	PSO		MGUG		MPUP		MXUX		MXUPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	1.23E-14	0	7.11E-15	0	7.11E-15	0	7.03E-15	0
GRIEWANK	0.00736214	48	0	100	0	100	0	100	7.40E-05	99
RASTRIGIN	104.351	0	0	100	0	100	0	100	0	100
ROSENBROCK	3.94E-31	98	6.10E-24	0	5.21E-23	0	1.31E-25	0	3.70E-31	99
SCHWEFEL	2382.29	0	5860.86	0	3849.72	0	2496.14	0	11.8529	93
COSINE	0.385717	8	0	100	0	100	0	100	0	100
EXPONENTIAL	1.25E-16	0	1.14E-16	3	1.78E-17	84	1.10E-16	1	0	100
LEVY	6.51724	3	1.25519	0	0.111015	15	0.312454	5	0.00179057	98
MICHALEWICZ	0.417773	0	0.530927	57	0.016026	98	0.238974	78	0	100
STEP	39.04	0	0	100	0	100	0	100	0	100
SCHAFFER	0	100	0.00252614	74	0	100	0	100	0	100
HOLDER	0	100	4.01E-07	89	1.61E-06	69	0	100	0	100
BEALE	0	100	0.0304832	96	0	100	0.0152414	98	0	100
SHUBERT	0	100	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100	0	100
TRID	3.29E-12	19	1.41E-11	1	1.62E-11	0	1.37E-11	1	1.46E-12	41
SPHERE	0	100	0	100	0	100	0	100	0	100
PARALLEL	0	100	0	100	0	100	0	100	0	100
ROTATED	0	100	0	100	0	100	0	100	0	100
CIGAR	0	100	0	100	0	100	0	100	0	100
BROWN	0	100	0	100	0	100	0	100	0	100
MULTIMOD	0	100	0	100	0	100	0	100	0	100
ZAKHAROV	3.35E-188	0	0	100	3.92E-210	0	0	100	0	100
EASOM	0	100	0	100	0	100	0	100	0	100
ผลรวม		1376		1720		1766		1783		2230

สำหรับตาราง 5.9 ได้ใช้พารามิเตอร์ตามตาราง 5.1 โดยตารางแสดงค่า MBF และ SR จากผลลัพธ์การค้นหาโดยรวมของ MGUG, MPUP, MXUX ดีกว่า PSO แต่สำหรับฟังก์ชันมัลติโมดอลบางฟังก์ชัน เช่นฟังก์ชัน SCHWEFEL ฟังก์ชัน ROSENBROCK ฟังก์ชัน LEVY ฟังก์ชัน HOLDER ฟังก์ชัน TRID ผลลัพธ์การค้นหาของ PSO ดีกว่า MGUG, MPUP, MXUX สำหรับฟังก์ชันยูนิโมดอล MGUG, MPUP, MXUX ผลลัพธ์การค้นหาดีกว่า PSO เพราะ MGUG, MPUP, MXUX มีความเร็วการลู่เข้าหากันของอนุภาคมากกว่า PSO จึงพบคำตอบได้เร็วกว่า PSO จากตารางผลการทดลองนี้สรุปได้ว่ากรการกลายพันธุ์แบบ CM (MGUG, MPUP, MXUX) ทำให้อนุภาคลู่เข้าหากันเร็วขึ้นซึ่งในปัญหามัลติโมดอลบางปัญหาอาจจะทำให้เกิดปัญหาการลู่เร็วเกินไปมากขึ้นกว่า PSO ปกติส่งผลให้ผลลัพธ์การค้นหาแยกว่า PSO ปกติ สำหรับปัญหาการลู่เร็วเกินไปสามารถแก้ไขได้โดยการกำหนดพารามิเตอร์ให้เหมาะสมกับปัญหาที่แก้ไขแต่การกำหนดพารามิเตอร์ให้เหมาะสมกับปัญหาที่แก้ไวนั้นเป็นเรื่องที่ทำได้ยาก ในขณะที่ฟังก์ชันมัลติโมดอลและฟังก์ชันยูนิโมดอลของ MXUPG ไม่มีฟังก์ชันไหนเลยที่ได้ผลลัพธ์การค้นหาแยกว่า PSO นอกจากนี้ MXUPG ได้ผลลัพธ์การค้นหาดีกว่า

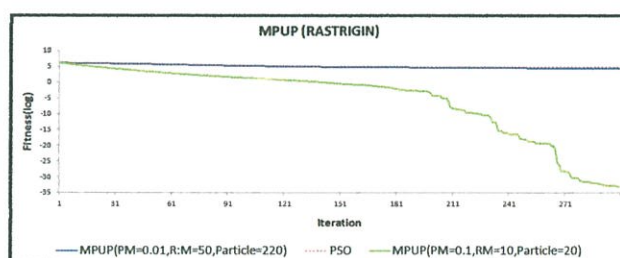
MGUG, MPUP, MXUX ด้วย เนื่องจาก MXUPG เพิ่มความหลากหลายให้กับประชากร ทำให้ลดหรือแก้ปัญหาค้างในจุดสุดสัมพัทธ์และ MXUPG ยังแทบที่จะไม่เพิ่มอัตราการลู่เข้าหากันของอนุภาคเมื่อเปรียบเทียบกับ PSO ดังนั้น MXUPG จึงแทบที่จะไม่ทำให้ประสบปัญหาการลู่เร็วเกินไปรุนแรงขึ้นเมื่อเปรียบเทียบกับ PSO ทำให้ MXUPG ไม่มีผลลัพธ์การค้นหาที่แย่กว่า PSO จากผลการทดลองในหัวข้อนี้สามารถสรุปได้ว่า MXUPG สามารถลดหรือแก้ปัญหาค้างในจุดสุดสัมพัทธ์ที่เกิดมาจากการกลายพันธุ์แบบ CM และสามารถเพิ่มประสิทธิภาพการค้นหาของ PSO โดยการเพิ่มความหลากหลายของประชากรซึ่งช่วยลดและแก้ปัญหาค้างในจุดสุดสัมพัทธ์

5.4.3 การพิจารณาปัญหาการกลายพันธุ์น้อยเกินไปของ CM

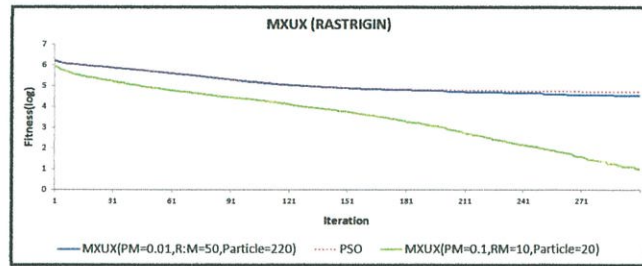
หัวข้อนี้ได้ทดลองการค้นหาค่าตอบด้วย PSO, MGUG, MPUP, MXUX, MXUPG กับฟังก์ชัน RASTRIGIN โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับกราฟ 5.23 ถึง 5.26 จำนวนรอบการค้นหาค่าตอบ 300 รอบ โดยกราฟแสดงค่า LMFx, LMFP, LMFG ของการค้นหาด้วย PSO, MGUG, MPUP, MXUX, MXUPG ของฟังก์ชัน RASTRIGIN ซึ่งถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอลตามลำดับ สำหรับตาราง 5.10 ได้ใช้พารามิเตอร์ตามตาราง 5.1 แต่จำนวนอนุภาคจะถูกกำหนดเท่ากับ 20 อนุภาคทุกขั้นตอนวิธี รอบของการกลายพันธุ์ถูกเปลี่ยนเป็นรอบของการค้นหาที่หารด้วย 2000 ลงตัว ความน่าจะเป็นในการกลายพันธุ์ถูกเปลี่ยนเป็น 0.00001 โดยตารางแสดงค่า MBF และ SR



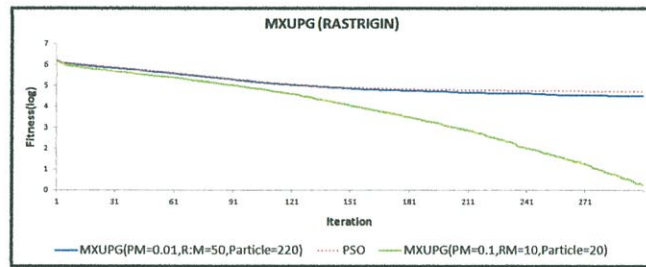
รูปที่ 5.23 ตัวอย่างปัญหาการกลายพันธุ์น้อยเกินไปของ MGUG



รูปที่ 5.24 ตัวอย่างปัญหาการกลายพันธุ์น้อยเกินไปของ MPUP



รูปที่ 5.25 ตัวอย่างปัญหาหลายพันธุ่น้อยเกินไปของ MXUX



รูปที่ 5.26 ตัวอย่างปัญหาหลายพันธุ่น้อยเกินไปของ MXUPG

จากกราฟ 5.23 ถึง 5.26 เส้นสีฟ้าแทน MGUG, MPUP, MXUX, MXUPG โดยพารามิเตอร์ของกราฟเหล่านี้กำหนดให้รอบของการกลายพันธุ์ถูกเปลี่ยนเป็นดังนี้ ถ้ารอบของการค้นหาด้วย 50 ลงตัว ความน่าจะเป็นในการกลายพันธุ์ถูกเปลี่ยนเป็น 0.01 จำนวนอนุภาคจะถูกกำหนดเท่ากับ 220 อนุภาค สำหรับเส้นประสีแดงแทน PSO ใช้พารามิเตอร์ตามตาราง 5.1 เส้นสีเขียวแทน MGUG, MPUP, MXUX, MXUPG โดยใช้พารามิเตอร์ตามตาราง 5.1

จากกราฟ 5.23 ถึง 5.26 จะเห็นว่าถ้าดำเนินการกลายพันธุ์น้อยเกินไป (เส้นกราฟสีฟ้า) การค้นหาก็ยังคงประสบปัญหาการติดในจุดสุดสัมพัทธ์เช่นเดียวกับ PSO (เส้นกราฟสีแดง) โดยเส้นกราฟทั้งสองมีค่าใกล้เคียงกันมาก ในขณะที่การกลายพันธุ์ที่เหมาะสม สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ โดยสังเกตจากเส้นกราฟที่มีค่าลดลงเรื่อยๆ ดังนั้นการกลายพันธุ์น้อยเกินไปนั้นไม่สามารถช่วยแก้ปัญหาการติดในจุดสุดสัมพัทธ์

จากตารางผลการทดลอง 5.10 แสดงให้เห็นว่าถ้ากำหนดให้ดำเนินการกลายพันธุ์น้อยเกินไป ผลลัพธ์ที่ได้จากการค้นหานั้นจะไม่ค่อยแตกต่างผลลัพธ์ที่ได้จากการค้นหาของ PSO เนื่องจากการกลายพันธุ์ที่เพิ่มเข้าไปไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์

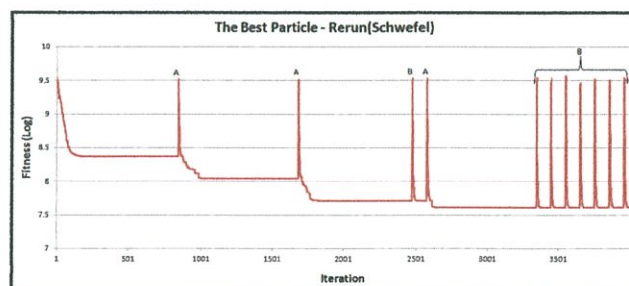
สำหรับปัญหาหลายพันธุ่น้อยเกินไปเกิดขึ้นได้ยาก เพราะปกติการกลายพันธุ์แบบ CM จะดำเนินการหลังจากการดำเนินการค้นหาด้วย PSO ทุกครั้ง ทำให้อย่างน้อยการค้นหาคำตอบหนึ่งรอบจะต้องดำเนินการกลายพันธุ์อย่างน้อยหนึ่งครั้ง ดังนั้นปัญหาหลายพันธุ่น้อยเกินไปจะเกิดขึ้นกับ CM จึงเป็นไปได้ยาก ยกเว้นผู้ออกแบบขั้นตอนวิธีตั้งใจจะให้เกิดขึ้นโดยกำหนดให้ดำเนินการกลายพันธุ์เพียงบางรอบการค้นหาคำตอบเท่านั้น จึงจะทำให้เกิดปัญหานี้ขึ้นซึ่งปัญหานี้ MXUPG ก็ไม่สามารถแก้ไขได้ แต่จากตารางผลการทดลอง 5.10 แสดงให้เห็นว่า MXUPG ก็ยังคงได้ผลลัพธ์โดยรวมดีที่สุด เนื่องจากปัญหานี้จะไม่เกิดขึ้นกับการกลายพันธุ์แบบ CM ที่ทำงานปกติดังนั้นปัญหาหลายพันธุ่น้อยเกินไปจึงไม่ใช่ปัญหาสำคัญที่ต้องแก้ไข

ตารางที่ 5.10 ผลการทดลองปัญหาหลายพันฐานน้อยเกินไปสำหรับ CM

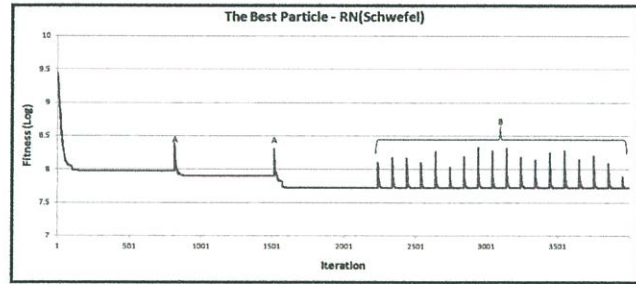
ALGORITHM	PSO		MGUG		MPUP		MXUX		MXUPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	5.33599	0	5.43604	0	5.70117	0	5.56861	0	4.92439	0
GRIEWANK	0.121359	1	0.0930537	3	0.169777	8	0.0795961	6	0.0980122	5
RASTRIGIN	157.004	0	159.054	0	155.89	0	160.576	0	156.158	0
ROSENBROCK	1.55E-22	0	2.95E-23	0	1.90E-23	0	1.67E-21	0	2.90E-24	0
SCHWEFEL	3876.31	0	4076.19	0	3891.54	0	3768.54	0	3705.75	0
COSINE	1.72662	0	1.7495	0	1.65519	0	1.74052	0	1.67857	0
EXPONENTIAL	3.58E-14	0	1.03E-14	0	4.66E-14	0	1.86E-14	0	2.82E-15	0
LEVY	26.588	0	26.9821	0	27.1755	0	27.1325	0	24.9513	0
MICHALEWICZ	1.17971	0	1.24849	0	1.19352	0	1.21587	0	1.16486	0
STEP	7498.82	0	6804.53	0	6141.42	0	6770.6	0	6761.1	0
SCHAFFER	0.00495515	48	0.00470474	51	0.00408068	57	0.00505229	47	0.00417784	57
HOLDER	1.15E-06	64	1.41E-06	75	1.43E-06	70	4.36E-06	74	1.47E-06	76
BEALE	0.0457273	94	0.022863	97	0.053346	93	0.0381056	95	0.0228629	97
SHUBERT	0	100	0	100	0	100	0	100	0	100
GOLDSTEIN	0.81	99	0	100	0	100	0.81	99	0	100
TRID	1.52E-11	0	1.48E-11	1	1.49E-11	1	1.49E-11	0	1.40E-11	1
SPHERE	9.30E-162	0	2.85E-160	0	5.16E-162	0	3.87E-164	0	9.92E-161	0
PARALLEL	5.51E-158	0	1.96E-160	0	3.06E-159	0	2.46E-160	0	1.39E-159	0
ROTATED	5.97E-162	0	6.07E-158	0	1.01E-166	0	1.65E-161	0	2.93E-168	0
CIGAR	3.45E-157	0	4.98E-154	0	2.04E-154	0	8.42E-155	0	9.90E-155	0
BROWN	3.40E-163	0	3.35E-158	0	7.13E-158	0	1.88E-154	0	1.19E-155	0
MULTIMOD	1.93895	0	1.18E-08	0	5.99E-19	0	2.28397	0	9.49E-06	0
ZAKHAROV	4.85E-62	0	4.48E-64	0	8.82E-66	0	5.01E-65	0	1.26E-63	0
EASOM	0	100	0	100	0	100	0	100	0	100
ผลรวม		506		527		529		521		536

5.5 การทดลองการประยุกต์ใช้รีรันและรีโพซิชั่น

ในหัวข้อนี้จะพิจารณาถึงความสามารถในการหาค้นหาค่าตอบฟังก์ชันมาตรฐานด้วย PSO รีรันและรีโพซิชั่น ในการทดลองนี้ได้เลือกใช้ RN เป็นตัวแทนของรีโพซิชั่น เพราะทั้ง RN และรีรัน ไม่มีขั้นตอนการรีเซ็ตทั้ง GBEST และ PBEST เพื่อวิเคราะห์ผลกระทบจากรีรันและรีโพซิชั่น โดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1 สำหรับกราฟ 5.27 และ 5.28 คือการทดลองรีโพซิชั่นและรีรันกับฟังก์ชัน SCHWEFEL ซึ่งถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอล โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่จำนวนรอบการค้นหาค่าตอบ 4000 รอบ และจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้ง เพราะแต่ละรอบของการทดลองการรีเซ็ตเกิดขึ้นไม่พร้อมกัน ถ้าเป็นค่าเฉลี่ยจะเกิดการรวมค่าแล้วหักล้างกันทำให้ไม่พบผลของการรีเซ็ต โดยกราฟทั้งสองแสดงค่า LBP ของการค้นหาค่าด้วย RERUN และ RN ของฟังก์ชัน SCHWEFEL



รูปที่ 5.27 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาค่าตอบด้วย RERUN



รูปที่ 5.28 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RN

จากกราฟ 5.27 และ 5.28 ช่วงแรกของการค้นหาเส้นกราฟจะลดลงแล้วคงที่ เมื่อเส้นกราฟคงที่เป็นระยะเวลาจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีรันหรือรีโพซิชั่น หลังจากดำเนินการรีรันหรือรีโพซิชั่นเส้นกราฟจะมีค่าเพิ่มเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุด ไปอยู่ในตำแหน่งที่แย่ลงกว่าเดิม หลังจากนั้นจะดำเนินการค้นหาด้วย PSO จึงทำให้เส้นกราฟลดลง ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้ค่าตอบนั้นดีขึ้น ในกรณีที่การรีเซ็ตเริ่มต้นที่จุด A สังเกตได้จากกราฟที่จะลู่ลงไปมากกว่าก่อนการรีเซ็ต ทำให้รีรันหรือรีโพซิชั่นสามารถช่วยปรับปรุงคำตอบของ PSO ให้ดีขึ้น แต่อย่างไรก็ตามโดยส่วนมากแล้วอนุภาคจะลู่กลับเข้ามาหา GBEST เดิม โดยที่ไม่พบตำแหน่งที่ดีกว่า GBEST เดิมในกรณีที่การรีเซ็ตเริ่มต้นที่จุด B สังเกตได้จากกราฟที่จะลู่กลับมาเท่าเดิม

เมื่อสังเกตความสูงของเส้นกราฟหลังจากการรีเซ็ต จะพบว่า RN มีความสูงน้อยกว่า RERUN เนื่องจาก RN ได้เปลี่ยนแปลงตำแหน่งของอนุภาคเพียงบางส่วน จึงเหลือส่วนที่เป็นผลจากการค้นหาเดิมอยู่ทำให้ตำแหน่งของอนุภาคที่ดีที่สุดแย่ลงกว่าเดิมไม่มากนัก เส้นกราฟจึงสูงต่ำกว่าตอนเริ่มต้นค้นหา ในขณะที่ RERUN ทำการรีเซ็ตอนุภาคทั้งหมดทำให้ตำแหน่งของอนุภาคทั้งหมดถูกกำหนดใหม่หมด เส้นกราฟจึงสูงเทียบเท่ากับตอนเริ่มต้นค้นหา ทำให้ขอบเขตการค้นหาของ RERUN มากกว่า RN ซึ่งทำให้ต้องเสียเวลาในการลู่เข้าหากันของ RERUN มากกว่า RN

สำหรับตาราง 5.11 การทดลองรีโพซิชั่นและรีรันโดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1 โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหา 5000 รอบ โดยตารางแสดงค่า ARR และสำหรับตาราง 5.12 คือการทดลองรีโพซิชั่นและรีรันโดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1 โดยใช้พารามิเตอร์ตามตาราง 5.1 โดยตารางแสดงค่า MBF และ SR

จากตารางผลการทดลอง 5.11 สำหรับฟังก์ชันมัลติโมดอลจำนวนรอบของการรีเซ็ตของ RN มีจำนวนมากกว่า RERUN หมายความว่ารีโพซิชั่นมีโอกาสในการลองค้นหามากกว่ารีรัน สำหรับฟังก์ชันยูนิโมดอลจำนวนรอบของการรีเซ็ตของทั้ง RN และ RERUN มีค่าเกือบเป็น 0 ทั้งหมด เพราะปัญหายูนิโมดอล PSO ไม่ได้ประสบปัญหาการติดในจุดสุดสัมพัทธ์ สำหรับฟังก์ชัน MULTIMOD ซึ่ง เป็นฟังก์ชันยูนิโมดอลการรีเซ็ตทั้งใน RN และ RERUN แต่ก็มีจำนวนที่ถูกรีเซ็ตน้อยมาก สาเหตุที่เกิดการรีเซ็ต เพราะการกำหนดค่าที่กำหนดรอบของการรีเซ็ตน้อยเกินไปจึงทำให้เกิดการรีเซ็ตบางในบางครั้ง โดยผลจากการรีเซ็ตอาจจะทำให้การค้นหาพบคำตอบช้าลงเล็กน้อย

ตารางที่ 5.11 ผลของ ARR ของฟังก์ชันมาตรฐานด้วย RERUN และ REPOSITION

ALGORITHM	RE-RUN	RE-POSITION (RN)
FUNCTION	ARR	ARR
ACKLEY	1.8025	3.5425
GRIEWANK	2.055	4.03
RASTRIGIN	0.475	2.0025
ROSENBROCK	1.625	3.2525
SCHWEFEL	2.0925	2.6325
COSINE	2.075	4.04
EXPONENTIAL	2.1075	4.2275
LEVY	2.005	2.78
MICHALEWICZ	2.225	3.5175
STEP	2.1	4.2625
SCHAFFER	2.3725	4.745
HOLDER	2.3125	4.615
BEALE	2.3	4.65
SHUBERT	2.35	4.6925
GOLDSTEIN	2.32	4.65
TRID	2.165	4.3375
SPHERE	0	0
PARALLEL	0	0
ROTATED	0	0
CIGAR	0	0
BROWN	0	0
MULTIMOD	0.18	0.33
ZAKHAROV	0	0
EASOM	0	0
ผลรวม	32.5625	62.3075

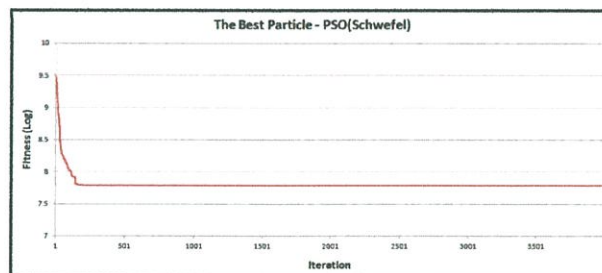
ตารางที่ 5.12 ผลการค้นหาคำตอบของฟังก์ชันคณิตศาสตร์ด้วย PSO, RERUN และ REPOSITION

ALGORITHM	PSO		RE-RUN		RE-POSITION (RN)	
	MBF	SR	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	0.0791055	0	7.74E-15	0
GRIEWANK	0.00736214	48	0.00927841	49	0.000246494	97
RASTRIGIN	104.351	0	99.0081	0	0	100
ROSENBROCK	3.94E-31	98	2.71E-31	99	2.96E-31	99
SCHWEFEL	2382.29	0	2320.7	0	1801.33	0
COSINE	0.385717	8	0.357638	10	0	100
EXPONENTIAL	1.25E-16	0	1.19E-16	0	1.10E-16	1
LEVY	6.51724	3	7.11549	4	0	100
MICHALEWICZ	0.417773	0	0.453917	2	0.0179453	98
STEP	39.04	0	43.54	0	0	100
SCHAFFER	0	100	0	100	0	100
HOLDER	0	100	0	100	0	100
BEALE	0	100	0	100	0	100
SHUBERT	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100
TRID	3.29E-12	19	3.82E-12	19	3.49E-12	14
SPHERE	0	100	0	100	0	100
PARALLEL	0	100	0	100	0	100
ROTATED	0	100	0	100	0	100
CIGAR	0	100	0	100	0	100
BROWN	0	100	0	100	0	100
MULTIMOD	0	100	0	100	0	100
ZAKHAROV	3.35E-188	0	2.11E-188	0	1.17E-188	0
EASOM	0	100	0	100	0	100
ผลรวม		1376		1383		1909

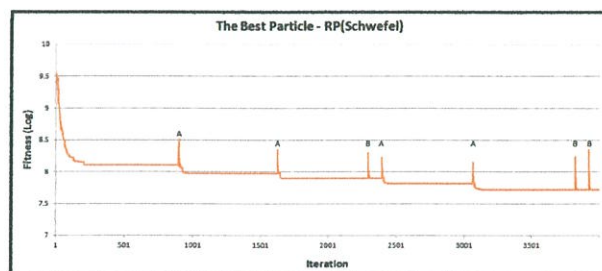
จากตารางผลการทดลอง 5.12 สำหรับฟังก์ชันมัลติโมดอล RERUN และ RN ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า PSO และฟังก์ชัน GRIEWANK, COSINE, LEVY จำนวนการพบคำตอบของ RN และ RERUN มีจำนวนที่มากกว่า PSO เนื่องจาก RN และ RERUN ได้ดำเนินการรีเซ็ตอนุภาคจึงมีโอกาสได้ลองค้นหาใหม่อีกครั้งแทนที่จะติดในจุดสุดสัมพัทธ์ จึงมีโอกาสในการลองค้นหาคำตอบมากกว่า PSO สำหรับฟังก์ชัน RASTRIGIN, MICHALEWICZ, STEP ทั้งสามฟังก์ชัน PSO ไม่สามารถหาคำตอบพบได้เลยเช่นเดียวกัน RERUN ก็ไม่สามารถหาคำตอบพบเช่นกัน แต่ RN สามารถหาคำตอบพบที่เป็นเช่นนี้เพราะ RN ไม่ได้แค่เพิ่มโอกาสให้การลองค้นหาคำตอบใหม่เพียงอย่างเดียว แต่ RN ได้ดำเนินการกลายพันธุ์กับตำแหน่งของอนุภาค ผลจากการกลายพันธุ์อาจจะทำให้อนุภาคมีโอกาสกระจายตัวไปยังตำแหน่งที่ PSO ไม่สามารถค้นหาพบ สำหรับฟังก์ชันยูนิโมดอลทั้ง PSO, RN และ RERUN ผลลัพธ์ที่ได้โดยรวมไม่แตกต่างกัน เพราะ RN และ RERUN แทบที่จะไม่ได้ดำเนินการรีเซ็ตหรือดำเนินการน้อยมากการค้นหาทั้งหมดจึงเป็นแบบ PSO ปกติ

5.6 การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพซิชั่น

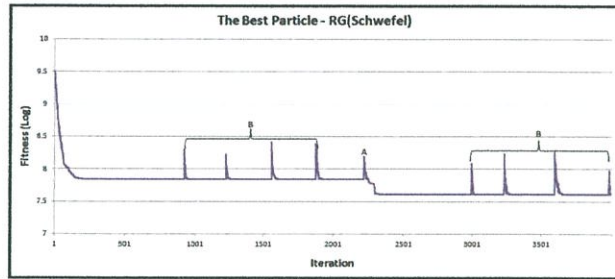
ในหัวข้อนี้จะพิจารณาถึงความสามารถในการค้นหาคำตอบจากฟังก์ชันมาตรฐานด้วย PSO, RG, RP, RN, RPG เพื่อแสดงให้เห็นผลกระทบจากการรีเซ็ต GBEST และ PBEST ของรีโพซิชั่น จึงได้ทำการทดลองโดยนำรีโพซิชั่นประเภทที่มีการรีเซ็ต GBEST เท่านั้น (RG) รีเซ็ต PBEST เท่านั้น (RP) ไม่รีเซ็ต GBEST กับ PBEST (RN) และรีเซ็ตทั้ง GBEST กับ PBEST (RPG) มาเปรียบเทียบกับโดยทดสอบกับฟังก์ชันมาตรฐานในตาราง ก.1 สำหรับกราฟ 5.29 ถึง 5.32 เป็นการทดลองรีโพซิชั่นกับฟังก์ชัน SCHWEFEL ซึ่งถูกเลือกเป็นตัวแทนของปัญหาโมดัล โดยใช้พารามิเตอร์ตามตาราง 5.1 แต่จำนวนรอบการค้นหาคำตอบ 4000 รอบ และจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้งเพื่อให้เห็นผลการรีเซ็ต โดยกราฟทั้งสี่แสดงค่า LBP ของการค้นหาด้วย PSO, RG, RP, RPG ของฟังก์ชัน SCHWEFEL



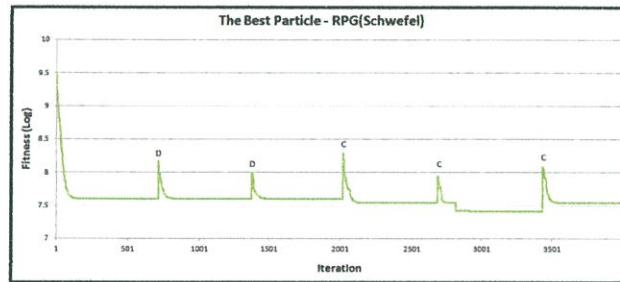
รูปที่ 5.29 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย PSO



รูปที่ 5.30 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย RP



รูปที่ 5.31 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาค่าตอบด้วย RG



รูปที่ 5.32 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาค่าตอบด้วย RPG

จากกราฟ 5.29 ช่วงแรกเส้นกราฟจะมีค่าลดลง เนื่องจากผลจากการค้นหาด้วย PSO และเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้นเส้นกราฟจะคงที่ตลอดจนสิ้นสุดการค้นหา สำหรับกราฟ 5.28, 5.30, 5.31 ช่วงแรกของการค้นหาเส้นกราฟจะลดลงแล้วคงที่ เมื่อเส้นกราฟคงที่เป็นระยะเวลาอันยาวนานจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีโพรซิชั่นและหลังจากดำเนินการรีโพรซิชั่น เส้นกราฟจะมีค่าเพิ่มเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่กว่าเดิม หลังจากนั้นจะดำเนินการค้นหาด้วย PSO จึงทำให้เส้นกราฟมีค่าลดลงซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้ค่าตอบนั้นดีขึ้นซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นที่จุด A เส้นกราฟหลังจากการรีเซ็ตจะถูกลงไปมากกว่าก่อนการรีเซ็ตทำให้สามารถช่วยในการปรับปรุงค่าตอบ แต่อย่างไรก็ตามโดยส่วนมากแล้วอนุภาคจะถูกลบเข้ามาหา GBEST เดิมโดยที่ไม่พบตำแหน่งที่ดีกว่า GBEST เดิมซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นที่จุด B หลังจากการรีเซ็ตเส้นกราฟถูกลงกลับมาเท่าเดิม

สำหรับจากกราฟ 5.32 ช่วงแรกของการค้นหาด้วย PSO เส้นกราฟจะลดลงและเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น เส้นกราฟจะคงที่เป็นระยะเวลาอันยาวนานจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีโพรซิชั่นหลังจากดำเนินการรีโพรซิชั่นเส้นกราฟจะมีค่าเพิ่มเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่กว่าเดิม ซึ่งในบางครั้งอนุภาคก็ถูกลบเข้ามาหาแล้วค้นหาในบริเวณใหม่ ซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นที่จุด C เส้นกราฟหลังจากการรีเซ็ตมีค่าลดลงแล้วคงที่ในระดับที่ต่างจากเส้นกราฟก่อนการรีเซ็ต แต่ในบางครั้งอนุภาคก็ถูกลบเข้ามาหาแล้วค้นหาในบริเวณเดิม ซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นที่จุด D เส้นกราฟหลังจากการรีเซ็ตมีค่าลดลงแล้วคงที่ในระดับที่เท่าเดิม

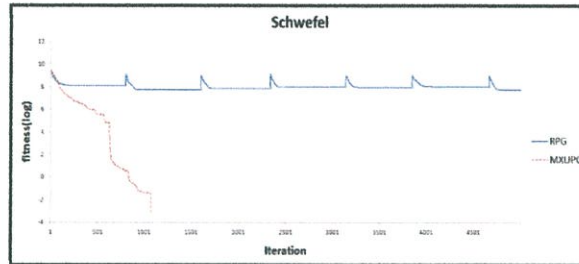
ตารางที่ 5.13 ผลการค้นหาคำตอบของฟังก์ชันมาตรฐานด้วย PSO, RG, RP, RN, RPG

ALGORITHM	PSO		RN		RP		RG		RPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	7.74E-15	0	7.18E-15	0	5.58E-15	0	0	100
GRIEWANK	0.00736214	48	0.000246494	97	1.11E-18	99	0.000271106	97	0	100
RASTRIGIN	104.351	0	0	100	0	100	0	100	0	100
ROSENBROCK	3.94E-31	98	2.96E-31	99	7.77E-31	93	0	100	0	100
SCHWEFEL	2382.29	0	1801.33	0	1661.44	0	1860.55	0	1472.22	0
COSINE	0.385717	8	0	100	0	100	0	100	0	100
EXPONENTIAL	1.25E-16	0	1.10E-16	1	1.07E-16	4	3.66E-17	67	0	100
LEVY	6.51724	3	0	100	0	100	0	100	0	100
MICHALEWICZ	0.417773	0	0.0179453	98	0.0593377	93	0.0551717	94	0.00996393	99
STEP	39.04	0	0	100	0	100	0	100	0	100
SCHAFFER	0	100	0	100	0	100	0	100	9.72E-05	99
HOLDER	0	100	0	100	0	100	0	100	0	100
BEALE	0	100	0	100	0	100	0	100	0	100
SHUBERT	0	100	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100	0	100
TRID	3.29E-12	19	3.49E-12	14	4.45E-12	7	1.09E-13	88	0	100
SPHERE	0	100	0	100	0	100	0	100	0	100
PARALLEL	0	100	0	100	0	100	0	100	0	100
ROTATED	0	100	0	100	0	100	0	100	0	100
CIGAR	0	100	0	100	0	100	0	100	0	100
BROWN	0	100	0	100	0	100	0	100	0	100
MULTIMOD	0	100	0	100	0	100	0	100	0	100
ZAKHAROV	3.35E-188	0	3.36E-187	0	3.14E-187	0	9.68E-188	0	3.36E-187	0
EASOM	0	100	0	100	0	100	0	100	0	100
ผลรวม		1376		1909		1896		2046		2198

สำหรับตาราง 5.13 ได้ใช้พารามิเตอร์ตามตาราง 5.1 โดยตารางแสดงค่า MBF และ SR จากผลการทดลองตารางนี้ สำหรับฟังก์ชันมัลติโมดอล RPG ได้ผลลัพธ์ที่ดีที่สุดเมื่อเปรียบเทียบกับ PSO, RP, RG, RN เพราะการค้นหาคำตอบด้วย RPG นั้นอนุภาคสามารถมีโอกาสเข้าไปค้นหาคำตอบในพื้นที่ใหม่ซึ่งแตกต่างจาก RN, RP, RG ที่ลู่เข้าหาพื้นที่เดิม ทำให้ RPG มีขอบเขตการค้นหาคำตอบที่กว้างกว่า RN, RP, RG ดังนั้น RPG จึงปรับปรุงประสิทธิภาพการค้นหาคำตอบของ PSO ได้มากที่สุด สำหรับฟังก์ชันยูนิโมดอลทั้ง PSO, RN, RP, RG, RPG ผลลัพธ์ที่ได้โดยรวมไม่แตกต่างกัน เพราะ RN, RP, RG, RPG ไม่ได้ดำเนินการรีเซ็ตหรือดำเนินการน้อยมาก ดังนั้นการค้นหาคำตอบทั้งหมดจึงเป็นแบบ PSO

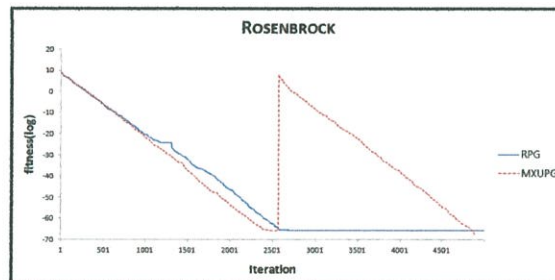
5.7 การทดลองผลกระทบจากการกลายพันธุ์และรีโพซิชัน

ในหัวข้อนี้จะพิจารณาถึงความสามารถในการค้นหาคำตอบจากฟังก์ชันมาตรฐานด้วย MXUPG และ RPG เพื่อแสดงให้เห็นผลกระทบจากปัญหาขอบเขตการกลายพันธุ์ของ MXUPG และปัญหาขาดความหลากหลายของ RPG สำหรับกราฟ 5.33 คือการทดลอง MXUPG และ RPG กับฟังก์ชัน SCHWEFEL ถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอล สำหรับกราฟ 5.34 คือการทดลอง MXUPG และ RPG กับฟังก์ชัน ROSENBROCK ถูกเลือกเป็นตัวแทนของปัญหามัลติโมดอล โดยทั้งสองกราฟใช้พารามิเตอร์ตามตาราง 5.1 แต่จำนวนรอบการค้นหาคำตอบ 5000 รอบและจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้งเพื่อให้เห็นผลของการรีเซ็ต โดยกราฟทั้งสองแสดงค่า LBP ของการค้นหาคำตอบด้วย MXUPG และ RPG ของฟังก์ชัน SCHWEFEL และฟังก์ชัน ROSENBROCK ตามลำดับ



รูปที่ 5.33 LBP ของฟังก์ชัน SCHWEFEL ที่ค้นหาคำตอบด้วย MXUPG และ RPG

จากกราฟ 5.33 เส้นกราฟของ MXUPG มีค่าลดลงสลับกับคงที่ในบางช่วงจนพบจุดสุดสัมบูรณ์ เพราะในขณะที่ติดอยู่ในจุดสุดสัมพัทธ์อนุภาคมีความหลากหลายที่มากทำให้อนุภาคกระจายตัวออกไปค้นหาคำตอบในบริเวณกว้างจึงมีโอกาสหลุดจากจุดสุดสัมพัทธ์และถ้าจุดสุดสัมบูรณ์อยู่ใกล้เคียงกับจุดสุดสัมพัทธ์ที่ติดอยู่ก็อาจจะทำให้พบจุดสุดสัมบูรณ์ได้ สำหรับเส้นกราฟของ RPG จะมีค่าลดลงแล้วคงที่เป็นระยะเวลาจนถึงขอบเขตที่กำหนดให้ดำเนินการรีเซ็ตก็จะทำการรีเซ็ตเส้นกราฟก็จะสูงขึ้นและลดลงอีก วนเช่นนี้ไปเรื่อยๆ โดยกราฟ RPG ไม่สามารถหาจุดสุดสัมบูรณ์พบเนื่องจาก RPG ประสบปัญหาขาดความหลากหลายโดยเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้นอนุภาคจะค้นหาในบริเวณที่แคบจึงไม่พบจุดสุดสัมบูรณ์



รูปที่ 5.34 LBP ของฟังก์ชัน ROSENBROCK ที่ค้นหาคำตอบด้วย MXUPG และ RPG

จากกราฟ 5.34 เส้นกราฟของ RPG มีค่าลดลงแล้วคงที่เป็นระยะเวลาจนถึงขอบเขตที่กำหนดให้ดำเนินการรีเซ็ตก็จะทำการรีเซ็ตเส้นกราฟก็จะสูงขึ้นและลดลง โดยหลังจากรีเซ็ตเส้นกราฟสามารถพบจุดสุดสัมบูรณ์ เพราะการรีเซ็ตทำให้อนุภาคกระจายตัวออกเป็นบริเวณกว้างแล้วทำให้อนุภาคคู่เข้าไปค้นหาในบริเวณใหม่ ซึ่งในกรณีคำตอบนั้นอยู่ไกลจากจุดสุดสัมพัทธ์ที่อนุภาคนั้นติดอยู่การรีเซ็ตจะทำให้อนุภาคมีโอกาสพบคำตอบ แต่สำหรับ MXUPG เส้นกราฟจะมีค่าลดลงในช่วงแรกแล้วต่อมาเส้นกราฟจะมีค่าคงที่ตลอดจนสิ้นสุดการค้นหาเหมือนกับ MXUPG นั้นติดอยู่ในจุดสุดสัมพัทธ์ เนื่องจาก MXUPG ประสบปัญหาขอบเขตการกลายพันธุ์ คือการกลายพันธุ์มีขอบเขตไม่กว้างพอที่จะครอบคลุมตำแหน่งของจุดสุดสัมบูรณ์หรือตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ที่อนุภาคนั้นติดอยู่จึงทำให้อนุภาคนั้นค้นหาเฉพาะบริเวณใดบริเวณหนึ่งเท่านั้น และไม่ออกไปค้นหาในบริเวณอื่นอีกจึงไม่สามารถพบจุดสุดสัมบูรณ์

5.8 การทดลองขั้นตอนวิธีที่นำเสนอกับฟังก์ชันมาตรฐาน

ตารางที่ 5.14 ผลการทดลองฟังก์ชันมาตรฐาน

ALGORITHM	PSO		HPSO		APSO		PSORF	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	0.0294295	0	5.54E-15	0	1.31234	0	0.0427766	0
GRIEWANK	0.00736214	48	0.00166348	98	0.0143407	28	0.00861458	50
RASTRIGIN	104.351	0	101.625	0	129.414	0	108.687	0
ROSENBROCK	3.94E-31	98	0	100	2.31E-29	53	4.35E-29	24
SCHWEFEL	2382.29	0	2452.96	2	2424.34	0	2389.59	0
COSINE	0.385717	8	0	100	0.994399	0	0.379806	6
EXPONENTIAL	1.25E-16	0	3.44E-17	69	3.01E-16	0	2.53E-16	0
LEVY	6.51724	3	0.430644	46	11.225	0	7.18407	2
MICHALEWICZ	0.417773	0	0.412666	2	0.689762	0	0.632524	0
STEP	39.04	0	3.55	14	922.57	0	81.45	0
SCHAFFER	0	100	0	100	0.000582955	94	0.00034627	88
HOLDER	0	100	0	100	0	100	0	100
BEALE	0	100	0	100	0	100	0	100
SHUBERT	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100
TRID	3.29E-12	19	3.02E-12	20	6.64E-13	75	1.33E-11	2
SPHERE	0	100	0	100	0	75	1.95E-246	1
PARALLEL	0	100	0	100	1.4822E-323	90	4.83E-238	2
ROTATED	0	100	0	100	1.97626E-323	93	2.06E-238	2
CIGAR	0	100	0	100	3.26083E-319	71	2.35E-231	1
BROWN	0	100	0	100	0	84	5.36E-257	0
MULTIMOD	0	100	0	100	4.38312	0	5.97E-230	0
ZAKHAROV	3.35E-188	0	4.73E-221	0	3.85371E-322	0	21.2043	0
EASOM	0	100	0	100	0	100	0	100
รวม		1376		1651		1163		678
ALGORITHM	PSORN		MXUPG		RPG		MRPSO	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	0.0842767	0	7.03E-15	0	0	100	0	100
GRIEWANK	0.00861511	44	7.40E-05	99	0	100	0	100
RASTRIGIN	100.401	0	0	100	0	100	0	100
ROSENBROCK	8.63E-31	95	3.70E-31	99	0	100	0	100
SCHWEFEL	2534.68	0	11.8529	93	1472.22	0	0	100
COSINE	0.347255	11	0	100	0	100	0	100
EXPONENTIAL	1.24E-16	0	0	100	0	100	0	100
LEVY	6.88109	3	0.00179057	98	0	100	0	100
MICHALEWICZ	0.426805	3	0	100	0.00996393	99	0	100
STEP	59.38	0	0	100	0	100	0	100
SCHAFFER	9.72E-05	99	0	100	9.72E-05	99	0	100
HOLDER	0	100	0	100	0	100	0	100
BEALE	0	100	0	100	0	100	0	100
SHUBERT	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100
TRID	2.95E-12	17	1.46E-12	41	0	100	0	100
SPHERE	0	100	0	100	0	100	0	100
PARALLEL	0	100	0	100	0	100	0	100
ROTATED	0	100	0	100	0	100	0	100
CIGAR	0	100	0	100	0	100	0	100
BROWN	0	100	0	100	0	100	0	100
MULTIMOD	0	100	0	100	3.36E-187	0	0	100
ZAKHAROV	4.46E-188	0	0	100	0	100	0	100
EASOM	0	100	0	100	0	100	0	100
รวม		1372		2230		2198		2400

ในหัวข้อนี้จะพิจารณาการค้นหาคำตอบด้วย PSO, MXUPG, RPG, APSO, HPSO, PSORE, PSORN และ MRPSO (ซึ่งเป็นวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ) กับฟังก์ชันมาตรฐาน เพื่อวิเคราะห์ประสิทธิภาพของการค้นหาคำตอบและความน่าเชื่อถือของขั้นตอนวิธีที่นำเสนอกับการแก้ปัญหาฟังก์ชันมาตรฐาน นอกจากนี้เปรียบเทียบประสิทธิภาพของการค้นหาของขั้นตอนวิธีที่นำเสนอ กับ PSO ในงานวิจัยอื่นกับการแก้ปัญหาฟังก์ชันมาตรฐาน โดยใช้พารามิเตอร์ตามตาราง 5.1 สำหรับตาราง 5.14 แสดงค่า MBF และ SR ของการค้นหาคำตอบ

จากตารางผลการทดลอง 5.14 แสดงให้เห็นว่า MRPSO ได้รับคำตอบที่ดีกว่า PSO, HPSO, APSO, PSOM, PSODM, MXUPG และ RPG ดังนั้น MRPSO จึงได้ผลลัพธ์การค้นหาคำตอบที่ดีที่สุด โดย MRPSO สามารถหาคำตอบได้ครบทุกฟังก์ชันที่นำมาทดลอง นอกจากนี้ SR ของ MRPSO แสดงให้เห็นถึงความน่าเชื่อถือของขั้นตอนวิธีโดย MRPSO สามารถหาคำตอบได้ 100 เปอร์เซ็นต์ทุกฟังก์ชัน สำหรับ MRPSO เกิดจากการรวมขั้นตอนวิธี RPG และ MXUPG ดังนั้นการประยุกต์ใช้ RPG และ MXUPG รวมกันสามารถเพิ่มประสิทธิภาพการค้นหาของ PSO ทั้งในฟังก์ชันยูนิโมดอลและฟังก์ชันมัลติโมดอล ซึ่งการประยุกต์ใช้ RPG และ MXUPG รวมกันจะช่วยลดข้อด้อยของกันและกัน จึงได้รับผลลัพธ์การค้นหาคำตอบที่ดีกว่าการประยุกต์ใช้เพียงขั้นตอนวิธีเดียว โดยสาเหตุที่ MRPSO สามารถหาคำตอบได้มีประสิทธิภาพมากกว่าขั้นตอนวิธี RPG และ MXUPG เพราะ MRPSO ได้แก้ปัญหาข้อเสียของกันและกันโดย MXUPG แก้ปัญหาขาดความหลากหลายของ RPG ในขณะที่ RPG แก้ปัญหาขอบเขตการกลายพันธุ์ของ MXUPG

5.9 การทดลองขั้นตอนวิธีที่นำเสนอกับปัญหาถุงเป้หลายมิติ

ในหัวข้อนี้จะพิจารณาการค้นหาคำตอบด้วย BPSO, GPBPSO, MBPSO, MXUPG, RPG และ MRPSO (ซึ่งเป็นวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ) กับปัญหาถุงเป้หลายมิติ เพื่อวิเคราะห์ประสิทธิภาพของการค้นหาคำตอบและความน่าเชื่อถือของขั้นตอนวิธีที่นำเสนอกับการแก้ปัญหาในรูปแบบไบนารี นอกจากนี้เปรียบเทียบประสิทธิภาพของการค้นหาของขั้นตอนวิธีที่นำเสนอ กับ BPSO ในงานวิจัยอื่นกับการแก้ปัญหาถุงเป้หลายมิติ โดยใช้พารามิเตอร์ตามตาราง 5.2 สำหรับตาราง 5.15 แสดงค่า MBF และ SR ของการค้นหาคำตอบ

การทดลองกับฟังก์ชันมาตรฐานเป็นการทดลองเพื่ออธิบายแนวความคิดที่มาของ MRPSO แต่การทดลองกับปัญหาถุงเป้หลายมิติ เพื่อแสดงการนำ MRPSO ประยุกต์ใช้งานกับปัญหาในรูปแบบไบนารีแล้วผลลัพธ์ในการค้นหาคำตอบยังคงมีประสิทธิภาพเหมือนกับการทดลองกับฟังก์ชันมาตรฐานหรือไม่ เนื่องจากปัญหาฟังก์ชันมาตรฐานเป็นปัญหาที่มีลักษณะเป็นเวกเตอร์ที่มีทั้งทิศทางและขนาด ทำให้อ้างอิงสมบัติและทฤษฎีต่างๆ ได้ง่าย เพราะทฤษฎีที่กล่าวมาทั้งหมดนั้นได้มีแนวคิดเริ่มต้นจากมุมมองของเวกเตอร์ที่มีทั้งทิศทางและขนาด โดย PSO ได้แนวคิดจากการที่อนุภาคเคลื่อนที่เข้าหากันโดยพิจารณาจากพื้นที่โดยรอบข้างซึ่งลักษณะของปัญหาเป็นพื้นผิว การพิจารณาพื้นที่โดยรอบก็จะช่วยในการคาดเดาค่าตำแหน่งของคำตอบได้จึงสามารถวิเคราะห์ปัญหานั้นได้อย่างมีประสิทธิภาพ

ตารางที่ 5.15 ผลการทดลองปัญหาถูกแก้หลายมิติ

ALGORITHM	BPSO		GPBPSO		MBPSO		MXUPG		RPG		MRPSO	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
SENTO1	7767.82	75	7772	100	7770.88	98	7771.72	98	7770.46	89	7772	100
SENTO2	8716.29	37	8722	100	8719.97	74	8719.64	76	8721.7	94	8722	100
WEING1	141278	100	141278	100	141278	100	141278	100	141278	100	141278	100
WEING2	130883	100	130883	100	130883	100	130883	100	130883	100	130883	100
WEING3	95643.7	91	95676	98	95661.6	98	95676.5	99	95677	100	95677	100
WEING4	119337	100	119337	100	119337	100	119337	100	119337	100	119337	100
WEING5	98796	100	98796	100	98796	100	98796	100	98796	100	98796	100
WEING6	130611	97	130623	100	130623	100	130623	100	130623	100	130623	100
WEING7	1.10E+06	9	1095380	0	1095370	27	1095300	8	1095440	95	1095450	100
WEING8	621019	13	620594	0	621366	14	624319	100	623284	68	624319	100
WEISH01	4554	100	4554	100	4554	100	4554	100	4554	100	4554	100
WEISH02	4535.3	86	4536	100	4536	100	4536	100	4535.25	85	4536	100
WEISH03	4115	100	4115	100	4115	100	4115	100	4115	100	4115	100
WEISH04	4561	100	4561	100	4561	100	4561	100	4561	100	4561	100
WEISH05	4514	100	4514	100	4514	100	4514	100	4514	100	4514	100
WEISH06	5553.24	78	5549.72	44	5556.46	97	5556.87	99	5557	100	5557	100
WEISH07	5567	100	5567	100	5567	100	5567	100	5567	100	5567	100
WEISH08	5604.94	97	5605	100	5605	100	5605	100	5605	100	5605	100
WEISH09	5246	100	5246	100	5246	100	5246	100	5246	100	5246	100
WEISH10	6339	100	6339	100	6339	100	6339	100	6339	100	6339	100
WEISH11	5643	100	5643	100	5643	100	5643	100	5643	100	5643	100
WEISH12	6339	100	6339	100	6339	100	6339	100	6339	100	6339	100
WEISH13	6159	100	6159	100	6159	100	6159	100	6159	100	6159	100
WEISH14	6954	100	6954	100	6954	100	6954	100	6954	100	6954	100
WEISH15	7486	100	7486	100	7486	100	7486	100	7486	100	7486	100
WEISH16	7288.88	94	7289	100	7288.78	97	7288.96	98	7289	100	7289	100
WEISH17	8633	100	8633	100	8633	100	8633	100	8633	100	8633	100
WEISH18	9578.35	87	9580	100	9579.85	99	9580	100	9580	100	9580	100
WEISH19	7696.83	91	7695.14	78	7697.61	97	7698	100	7698	100	7698	100
WEISH20	9449.6	98	9450	100	9450	100	9450	100	9450	100	9450	100
WEISH21	9073.52	98	9074	100	9073.76	99	9074	100	9074	100	9074	100
WEISH22	8938.18	51	8930.44	8	8940.7	65	8943.58	81	8946.64	98	8947	100
WEISH23	8343.67	89	8343.94	98	8343.76	96	8343.94	98	8344	100	8344	100
WEISH24	10220	100	10220	100	10220	100	10220	100	10220	100	10220	100
WEISH25	9934.66	73	9939	100	9937.56	91	9938.01	93	9939	100	9939	100
WEISH26	9574.93	73	9584	100	9576.96	78	9578.24	82	9584	100	9584	100
WEISH27	9819	100	9819	100	9819	100	9819	100	9819	100	9819	100
WEISH28	9492	100	9492	100	9491.77	99	9492	100	9492	100	9492	100
WEISH29	9410	100	9410	100	9409.52	98	9410	100	9410	100	9410	100
WEISH30	11190.2	81	11191	100	11190.8	94	11190.8	94	11191	100	11191	100
PB1	3083.39	72	3090	100	3090	100	3090	100	3088.46	89	3090	100
PB2	3178.33	69	3186	100	3184.2	93	3185.82	99	3185.46	97	3186	100
PB4	95168	100	95168	100	95168	100	95168	100	95168	100	95168	100
PB5	2133.53	70	2137.64	96	2139	100	2139	100	2139	100	2139	100
PB6	775.58	97	776	100	776	100	776	100	776	100	776	100
PB7	1033.67	85	1035	100	1034.8	98	1034.91	99	1034.91	99	1035	100
HP1	3410.74	65	3404	0	3416.61	95	3418	100	3416.32	88	3418	100
HP2	3178.79	75	3186	100	3185.82	99	3186	100	3185.64	98	3186	100
ผลรวม		4151		4322		4506		4624		4700		4800

สำหรับปัญหาถุงเป้หลายมิติมีลักษณะเป็นการเลือกหรือไม่เลือกถ้ามองในรูปแบบเวกเตอร์จะเป็นเวกเตอร์หลายมิติที่ไม่มีทิศทางและไม่มีขนาด การคาดเดาตำแหน่งของคำตอบจึงไม่สามารถวิเคราะห์พื้นที่โดยรอบข้างได้ ดังนั้นแนวคิดต่างๆ ที่ได้กล่าวมานั้นอาจจะไม่มีประสิทธิภาพจึงจำเป็นต้องทดลองกับปัญหานี้เพื่อพิสูจน์ว่าทฤษฎีที่กล่าวมาข้างต้นนั้นยังคงมีประสิทธิภาพ ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้ทำการทดลองเปรียบเทียบผลลัพธ์ของ BPSO, GPBPSO, MBPSO, MXUPG, RPG และ MRPSO (ซึ่งเป็นวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ) กับปัญหาถุงเป้หลายมิติเพื่อแสดงว่าเมื่อนำ MRPSO ประยุกต์ใช้งานกับปัญหาในรูปแบบไบนารีแล้วผลลัพธ์ในการค้นหาคำตอบยังคงมีประสิทธิภาพ สำหรับขั้นตอนการนำ PSO ประยุกต์ใช้กับปัญหาถุงเป้หลายมิติได้เขียนอธิบายไว้ในภาคผนวก ข

จากตารางผลการทดลอง 5.15 แสดงให้เห็นว่า MRPSO ได้รับคำตอบที่ดีที่สุดเมื่อเปรียบเทียบกับ BPSO, GPBPSO, MBPSO, MXUPG, RPG โดย MRPSO สามารถหาคำตอบของปัญหาที่นำมาทดลองได้ครบทุกปัญหา นอกจากนี้ผลรวมของ SR ของ MRPSO แสดงให้เห็นว่า MRPSO มีความน่าเชื่อถือของ ขั้นตอนวิธีมากที่สุด เพราะสามารถหาคำตอบพบ 100 เปอร์เซ็นต์ได้ครบทุกปัญหา ดังนั้น MRPSO สามารถเพิ่มประสิทธิภาพการค้นหาและความน่าเชื่อถือของ BPSO ได้มากกว่า GPBPSO และ MBPSO และจากผลการทดลองนี้เป็นการยืนยันอีกครั้งว่าการประยุกต์ใช้ MXUPG ร่วมกับ RPG ได้ผลลัพธ์ดีกว่าการประยุกต์ใช้อย่างใดอย่างหนึ่งเพียงอย่างเดียว นอกจากนี้ผลการทดลองแสดงให้เห็นว่านอกจากปัญหาฟังก์ชันคณิตศาสตร์ MRPSO ยังคงมีประสิทธิภาพในการแก้ปัญหาถุงเป้หลายมิติซึ่งเป็นปัญหาในรูปแบบไบนารีด้วย จากผลการทดลองสรุปว่า MRPSO แม้ว่าจะถูกนำไปประยุกต์ใช้กับปัญหาในรูปแบบไบนารีผลลัพธ์ที่ได้ก็ยังมีประสิทธิภาพ

5.10 การทดลองขั้นตอนวิธีที่นำเสนอกับปัญหาการเดินทางของพนักงานขาย

ในหัวข้อนี้จะพิจารณาการค้นหาคำตอบด้วย PPSO, MPPSO, RPG, MXUPG และ MRPSO (ซึ่งเป็นวิธีที่วิทยานิพนธ์ฉบับนี้นำเสนอ) กับปัญหาการเดินทางของพนักงานขาย เพื่อวิเคราะห์ประสิทธิภาพของการค้นหาคำตอบและความน่าเชื่อถือของขั้นตอนวิธีที่นำเสนอกับการแก้ปัญหาในรูปแบบการสลับลำดับกันในพีชคณิต นอกจากนี้เปรียบเทียบประสิทธิภาพของการค้นหาของขั้นตอนวิธีที่นำเสนอกับ PPSO ในงานวิจัยอื่นกับการแก้ปัญหาการเดินทางของพนักงานขาย โดยใช้พารามิเตอร์ตามตาราง 5.3 สำหรับตาราง 5.16 แสดงค่า MBF และ SR ของการค้นหาคำตอบ

การทดลองกับฟังก์ชันมาตรฐานเป็นการทดลองเพื่ออธิบายแนวความคิดที่มาของ MRPSO แต่การทดลองกับปัญหาการเดินทางของพนักงานขาย เพื่อแสดงการนำ MRPSO ประยุกต์ใช้งานกับปัญหาในรูปแบบการสลับลำดับกันในพีชคณิตแล้วผลลัพธ์ในการค้นหาคำตอบยังคงมีประสิทธิภาพเหมือนกับการทดลองกับฟังก์ชันมาตรฐานและปัญหาถุงเป้หลายมิติหรือไม่

เช่นเดียวกับปัญหาถุงเป้หลายมิติ คือปัญหาการเดินทางของพนักงานขายเป็นเวกเตอร์หลายมิติที่ไม่มีทิศทางและไม่มีขนาด การคาดเดาตำแหน่งของคำตอบจึงไม่สามารถวิเคราะห์พื้นที่โดยรอบข้าง ดังนั้นแนวคิดต่างๆ ได้กล่าวมาอาจจะไม่มีประสิทธิภาพ จึงจำเป็นต้องทดลองเพื่อพิสูจน์ว่าทฤษฎีที่กล่าวมาข้างต้นนั้นยังคงมีประสิทธิภาพ ดังนั้นวิทยานิพนธ์ฉบับนี้จึงได้ทำการทดลองเปรียบเทียบผลลัพธ์ของ PPSO, MPPSO, MXUPG, RPG และ MRPSO กับปัญหาการเดินทางของพนักงานขาย เพื่อแสดงว่าเมื่อนำ MRPSO ประยุกต์ใช้งานกับปัญหาในรูปแบบการสลับลำดับกันในพีชคณิตแล้ว

ผลลัพธ์ในการค้นหาคำตอบยังคงมีประสิทธิภาพ สำหรับขั้นตอนการนำ PSO ประยุกต์ใช้กับปัญหาการเดินทางของพนักงานขายได้เขียนอธิบายไว้ในภาคผนวก ค

ตารางที่ 5.16 ผลการทดลองปัญหาการเดินทางของพนักงานขาย

ALGORITHM	PPSO		MPPSO		MXUPG		RPG		MRPSO	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
BURMA14	3327.55	65	3332.34	69	3323	100	3323	100	3323	100
ULYSSES16	6877.92	41	6901.26	17	6859	100	6859	100	6859	100
GR17	2096.3	69	2109.7	47	2085	100	2085	100	2085	100
GR21	2818.69	39	2967.16	14	2707	100	2707	100	2707	100
ULYSSES22	7122.33	6	7217.01	4	7032.38	77	7014.95	95	7013	100
FRI26	973.43	1	1009.94	2	949.24	30	943.44	58	937	100
BAYG29	1753.5	0	1833.78	0	1627.34	38	1635.58	2	1610	100
BAYS29	2099.31	1	2246.98	0	2032.56	23	2040.92	5	2020	100
DANTZIG42	823.7	0	730.3	0	708.3	4	725.2	0	699	10
SWISS42	1446	0	1356	1	1332.7	1	1319.8	2	1273	10
GR48	5946.4	0	5569.7	0	5183.8	2	5393.6	0	5046	10
EIL51	502.5	0	449.8	0	432.8	5	446.2	0	426	10
BERLIN52	8841.2	0	8205.3	0	7994.4	0	8155.4	0	7542	10
BRAZIL58	29096.6	0	27338.4	0	26107.6	0	27646.4	0	25395	10
ผลรวม		222		154		580		562		860

จากตารางผลการทดลอง 5.16 แสดงให้เห็นว่า MRPSO ได้รับคำตอบที่ดีที่สุดเมื่อเปรียบเทียบกับ PPSO, MPPSO, MXUPG, RPG โดย MRPSO สามารถหาคำตอบของปัญหาที่นำมาทดลองได้ครบพบทุกปัญหา นอกจากนี้ ผลรวมของ SR ของ MRPSO แสดงให้เห็นว่า MRPSO มีความน่าเชื่อถือของขั้นตอนวิธีมาก เพราะสามารถหาคำตอบพบ 100 เปอร์เซ็นต์ได้ครบพบทุกปัญหา ดังนั้น MRPSO สามารถเพิ่มประสิทธิภาพการค้นหาและความน่าเชื่อถือของ PPSO ได้มากกว่า MPPSO ผลการทดลองนี้เป็นการยืนยันอีกครั้งว่าการประยุกต์ใช้ MXUPG ร่วมกับ RPG ได้ผลลัพธ์ดีกว่าการประยุกต์ใช้อย่างใดอย่างหนึ่งเพียงอย่างเดียว นอกจากนี้ผลการทดลองแสดงให้เห็นว่านอกจากปัญหาฟังก์ชันคณิตศาสตร์และปัญหาถุงเป้หลายมิติ MRPSO ยังคงมีประสิทธิภาพในการแก้ปัญหาการเดินทางของพนักงานขายซึ่งเป็นปัญหาในรูปแบบการสลับลำดับกันในพีชคณิต จากผลการทดลองสรุปว่า MRPSO แม้ว่าจะถูกนำไปประยุกต์ใช้กับปัญหารูปแบบการสลับลำดับกันในพีชคณิตผลลัพธ์ที่ได้ก็ยังมีประสิทธิภาพ

บทที่ 6

สรุปผลการทดลองและข้อเสนอแนะ

6.1 สรุปผลการทดลอง

ขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO) เป็นขั้นตอนวิธีที่นิยมนำมาประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุด ทั้งในงานด้านการขนส่ง วิทยาศาสตร์ วิศวกรรม ธุรกิจ และอุตสาหกรรม เป็นต้น โดยมีงานวิจัยจำนวนมากเช่น งานวิจัย [119, 120] ได้นำ PSO มาประยุกต์ใช้ฝึกโครงข่ายประสาทและปรับพารามิเตอร์ของโครงข่ายประสาท สำหรับงานวิจัย [121 - 123] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาการแบ่งกลุ่มข้อมูล สำหรับงานวิจัย [124, 125] ได้นำ PSO มาประยุกต์ใช้แก้ปัญหาเรื่องขนส่ง เป็นต้น

เมื่อเปรียบเทียบ PSO กับขั้นตอนวิธีเชิงวิวัฒนาการพบว่า PSO สามารถแก้ปัญหาการหาจุดที่ดีที่สุดได้ดีกว่าหลายปัญหา โดย PSO มีความเร็วในการหาคำตอบมากกว่าและได้รับคำตอบจากการค้นหาที่ดีกว่า นอกจากนี้ PSO มีข้อดีอื่นอีกมากมายเช่น สามารถนำไปประยุกต์ใช้กับปริภูมิของคำตอบขนาดใหญ่ สามารถเขียนโปรแกรมหรือนำไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดได้ง่าย สามารถนำไปประยุกต์ใช้ปัญหาที่มีรูปแบบปริภูมิคำตอบถูกแทนด้วยจำนวนจริง สามารถนำไปประยุกต์ใช้ปัญหาที่มีรูปแบบปริภูมิคำตอบถูกแทนด้วยไบนารี สามารถนำไปประยุกต์ใช้ปัญหาที่มีรูปแบบปริภูมิคำตอบถูกแทนด้วยการสลับลำดับกันในพื้นที่คณิต การคำนวณของ PSO มีประสิทธิภาพที่ดี พารามิเตอร์ของ PSO มีจำนวนน้อยที่ต้องการการกำหนดค่า เป็นต้น จึงเป็นสาเหตุให้งานวิจัยจำนวนมากนำ PSO ไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุด

แต่อย่างไรก็ตาม PSO มีข้อเสียหลักคือปัญหาการลู่เร็วเกินไปหรือปัญหาการติดในจุดสุดสัมพัทธ์ ซึ่งทำให้ผลลัพธ์จากการค้นหาได้รับคำตอบไม่ดีเท่าที่ควรและเมื่อนำ PSO ไปประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดจะได้รับผลลัพธ์จากการค้นหาไม่ดีเท่าที่ควร เพื่อแก้หรือลดปัญหาการติดในจุดสุดสัมพัทธ์ วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการปรับปรุง PSO ให้มีประสิทธิภาพการค้นหาที่ดีขึ้น โดยวิธีการที่นำเสนอคือการประยุกต์การกลายพันธุ์และรีโพอิชชันกับวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (MRPSO) ซึ่งเป็นวิธีที่เกิดจากการประยุกต์นำการกลายพันธุ์กับตำแหน่งของอนุภาคและผลการกลายพันธุ์ใช้ปรับปรุง PBEST และ GBEST (MXUPG) ร่วมกับรีโพอิชชันที่รีเซ็ต PBEST และ GBEST (RPG) โดยทั้งสองเทคนิคนี้เมื่อประยุกต์ใช้ร่วมกันสามารถแก้หรือลดปัญหาการติดในจุดสุดสัมพัทธ์และเพิ่มประสิทธิภาพการค้นหาของ PSO ให้ได้รับคำตอบจากการค้นหาที่ดีขึ้น

สำหรับการนำการกลายพันธุ์ไปประยุกต์ใช้กับ PSO โดยปกติแล้วสามารถเพิ่มความหลากหลายให้กับประชากรจึงทำให้อนุภาคสามารถหลีกเลี่ยงหรือหลุดจากจุดสุดสัมพัทธ์ได้ดีกว่า PSO ปกติ เพราะได้เพิ่มขอบเขตการค้นหาคำตอบให้กว้างขึ้นในขณะที่ติดอยู่ในจุดสุดสัมพัทธ์ โดยขอบเขตการค้นหาที่กว้างขึ้นนี้ ถ้าเพียงพอที่จะทำให้อนุภาคหลุดออกจากจุดสุดสัมพัทธ์จะทำให้ได้รับคำตอบจากการค้นหาที่ดีกว่า PSO ปกติ แต่สิ่งที่จะต้องควรระวังคือถ้าหากกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป อาจทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบสำหรับในกรณีของการกลายพันธุ์ประเภทไม่มีการเปรียบเทียบ (NCM) และปัญหาการลู่เร็วเกินไปสำหรับในกรณีของการกลายพันธุ์ประเภทมีการเปรียบเทียบ (CM) ซึ่งทำให้ผลลัพธ์จากการค้นหาคำตอบแย่กว่า PSO ที่ไม่

ประยุกต์การกลายพันธุ์ ดังนั้นการนำการกลายพันธุ์ไปประยุกต์ใช้กับ PSO เพื่อแก้ปัญหาการหาจุดที่ดีที่สุดโดยทั่วไปจึงเป็นเรื่องยุ่งยาก เพราะต้องคอยระวังเรื่องพารามิเตอร์ที่ปรับจะไม่เหมาะสมกับปัญหาที่นำไปแก้จึงได้ผลลัพธ์ที่แย่กว่า PSO ที่ไม่ดำเนินการกลายพันธุ์

สำหรับวิทยานิพนธ์ฉบับนี้ได้เสนอวิธีการปรับปรุง CM ให้มีประสิทธิภาพการค้นหาที่ดีขึ้นโดยแก้หรือลดปัญหาการลู่เร็วเกินไป ซึ่งวิธีที่นำเสนอคือการกลายพันธุ์จะประยุกต์ใช้กับอนุภาค แต่ผลจากการกลายพันธุ์จะถูกนำมาปรับปรุง PBEST และ GBEST เท่านั้นซึ่งเรียกวิธีนี้ว่า MXUPG วิธีนี้จะเพิ่มความหลากหลายให้กับประชากร โดยแทบที่จะไม่เพิ่มปัญหาการลู่เร็วเกินไปเมื่อเปรียบเทียบกับ PSO ที่ไม่ประยุกต์การกลายพันธุ์ จากผลการทดลองในบทที่ 5 พบว่า MXUPG สามารถเพิ่มประสิทธิภาพการค้นหาของ PSO ให้ได้รับคำตอบที่ดีขึ้น ทั้งในปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ ปัญหาการเดินทางของพนักงานขาย

แต่อย่างไรก็ตาม MXUPG ก็มีโอกาที่จะติดอยู่ในจุดสุดสัมพัทธ์เช่นเดียวกับ PSO ถ้าหากขอบเขตของการกลายพันธุ์ไม่มากพอที่จะพบตำแหน่งที่ดีกว่าจุดสุดสัมพัทธ์ที่ติดอยู่ เรียกปัญหานี้ว่า ปัญหาขอบเขตการกลายพันธุ์ เพื่อแก้ปัญหานี้จึงได้นำวิธีการรีรันมาประยุกต์ใช้กับ MXUPG โดยเมื่อ MXUPG เกิดปัญหาการติดอยู่ในจุดสุดสัมพัทธ์ เพื่อให้อนุภาคเกิดการกระจายตัวเป็นบริเวณกว้างและมีโอกาสในการค้นหาใหม่อีกครั้ง เพราะรีรันมีขอบเขตการกระจายตัวที่กว้างกว่าขอบเขตการกระจายตัวของการกลายพันธุ์มาก จึงมีโอกาที่อนุภาคจะกระจายตัวออกไปพบคำตอบที่ดีกว่าจุดสุดสัมพัทธ์ที่ติดอยู่โดยที่การกลายพันธุ์นั้นไม่สามารถกระจายตัวออกไปพบ ทำให้อนุภาคหลุดออกจากจุดสุดสัมพัทธ์

แต่รีรันได้ดำเนินการเกินความจำเป็น เพราะทำให้อนุภาคกระจายตัวออกเป็นบริเวณกว้างเกินความจำเป็นทำให้สูญเสียเวลาในการลู่เข้าหากันของอนุภาคมากเกินไป ผลที่ตามมาคือรอบในการรีเซ็ตจะลดลง ทำให้สูญเสียโอกาสในการลองค้นหาคำตอบ เพื่อลดเวลาในการลู่เข้าหากันของอนุภาค วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีโพซิชั่นมาประยุกต์ใช้แทนรีรัน ซึ่งวิธีนี้สามารถกำหนดขอบเขตการกระจายตัวของอนุภาคได้ จึงทำให้สามารถกำหนดให้อนุภาคกระจายตัวออกจากกันเท่าที่จำเป็น ดังนั้นเวลาในการลู่เข้าหากันจึงลดลง ทำให้โอกาสในการลองค้นหาคำตอบจึงเพิ่มขึ้น ประสิทธิภาพการค้นหาคำตอบจึงดีขึ้นและได้รับคำตอบจากการค้นหาที่ดีขึ้น จากผลการทดลองในบทที่ 5 พบว่า รีโพซิชั่นมีจำนวนรอบในการรีเซ็ตมากกว่ารีรัน และผลลัพธ์จากการค้นหาคำตอบของรีโพซิชั่นได้รับคำตอบที่ดีกว่ารีรัน และ PSO ปกติ ทั้งในปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ ปัญหาการเดินทางของพนักงานขาย

แต่รีโพซิชั่นเองถ้าหากไม่ดำเนินการรีเซ็ต PBEST และ GBEST ด้วยหลักการทำงานของ PSO จะทำให้อนุภาคทั้งหมดนั้นลู่กลับมาหา GBEST ที่ติดอยู่ในจุดสุดสัมพัทธ์ การค้นหาก็จะเหมือนกับการค้นหาขอบบริเวณใดบริเวณหนึ่ง ซึ่งในความจริงแล้วการค้นหาควรจะกระจายตัวไปทั่วปริภูมิคำตอบ จึงจะมีโอกาสพบคำตอบที่ดีขึ้น เพื่อให้อนุภาคกระจายตัวไปค้นหาในบริเวณใหม่ รีโพซิชั่นต้องดำเนินการรีเซ็ต PBEST และ GBEST ซึ่งเรียกวิธีรีโพซิชั่นที่ดำเนินการรีเซ็ต PBEST และ GBEST ว่า RPG จากผลการทดลองในบทที่ 5 พบว่า RPG สามารถเพิ่มประสิทธิภาพการค้นหาของ PSO ให้ได้รับคำตอบที่ดีขึ้น และเมื่อเปรียบเทียบ RPG กับรีโพซิชั่นที่ไม่ดำเนินการรีเซ็ต PBEST และ GBEST พบว่า RPG ได้ผลลัพธ์การค้นหาคำตอบดีกว่าทั้งในปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ ปัญหาการเดินทางของพนักงานขาย

สำหรับการใช้ RPG เพียงอย่างเดียวนั้น ประชากรจะขาดความหลากหลายเมื่อติดอยู่ในจุดสุดสัมพัทธ์ อนุภาคจะค้นหาในบริเวณแคบจึงไม่สามารถหลุดจากจุดสุดสัมพัทธ์ ถึงแม้จะมีการรีเซ็ตก็ไม่สามารถทำให้พบคำตอบในกรณีที่คำตอบนั้นอยู่ใกล้กับบริเวณจุดสุดสัมพัทธ์ที่ติดอยู่ เรียกปัญหานี้ว่า ปัญหาขาดความหลากหลาย

เพื่อให้ประชากรมีความหลากหลายมากในขณะที่อนุภาคติดอยู่ในจุดสุดสัมพัทธ์ จึงได้นำ MXUPG มาประยุกต์ใช้งานร่วมกับ RPG เพราะ MXUPG สามารถเพิ่มความหลากหลายให้กับประชากรในขณะที่อนุภาคติดอยู่ในจุดสุดสัมพัทธ์ จึงแก้ปัญหาขาดความหลากหลายของ RPG ได้ เรียกวิธีที่ประยุกต์ใช้ร่วมกันนี้ว่า MRPSO จากการทดลองในบทที่ 5 พบว่าการประยุกต์ MXUPG ร่วมกับ RPG สามารถเพิ่มประสิทธิภาพการค้นหาและได้รับคำตอบที่ดีกว่าการประยุกต์ใช้ MXUPG หรือ RPG อย่างใดอย่างหนึ่ง และได้รับจึงคำตอบจากการค้นหาดีกว่า PSO ทั้งในปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ และปัญหาการเดินทางของพนักงานขาย

จากผลการทดลองในบทที่ 5 ในหัวข้อที่ 5.8, 5.9, 5.10 จะเห็นว่าวิธีที่วิทยานิพนธ์ฉบับนี้ นำเสนอ (MRPSO) ได้ผลลัพธ์ในการค้นหาคำตอบดีกว่าขั้นตอนวิธีในเอกสารอื่นที่นำมาเปรียบเทียบ โดยขั้นตอนวิธีเหล่านี้มีลักษณะใกล้เคียงกันขั้นตอนวิธีที่นำเสนอ ทั้งในปัญหาฟังก์ชันมาตรฐาน (PSO, APSO, HPSO, PSORE, PSORN) ปัญหาถุงเป้หลายมิติ (BPSO, GPBPSO, MBPSO) ปัญหาการเดินทางของพนักงานขาย (PPSO, MPPSO) จากผลการทดลองแสดงให้เห็นว่า MRPSO มีประสิทธิภาพในการค้นหาดีกว่า PSO, APSO, HPSO, PSORE, PSORN, BPSO, GPBPSO, MBPSO, PPSO และ MPPSO ทั้งในด้านคุณภาพของคำตอบที่ได้รับและความน่าเชื่อถือของขั้นตอนวิธี โดย MRPSO สามารถหาคำตอบพบในทุกปัญหาที่นำมาทดลองและหาคำตอบได้พบทุกครั้ง เนื่องจาก MRPSO สามารถลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO ได้ดีที่สุดเมื่อเปรียบเทียบกับ PSO ที่ถูกนำมาใช้ในการทดสอบเปรียบเทียบทั้งหมด จึงทำให้ MRPSO ได้รับผลลัพธ์การค้นหาคำตอบที่ดีที่สุด

6.2 ข้อเสนอแนะ

วิธีที่นำเสนอ (MRPSO) ได้ใช้หลักการทำงานของ PSO เป็นหลัก ดังนั้นข้อดีของ MRPSO จึงได้รับการสืบทอดมาจาก PSO เช่น พารามิเตอร์ของ MRPSO มีจำนวนน้อยที่ต้องการการกำหนดค่า ขั้นตอนการทำงานของ MRPSO มีรูปแบบไม่ซับซ้อน การคำนวณของ MRPSO มีประสิทธิภาพที่ดี MRPSO สามารถประยุกต์ใช้กับจำนวนจริง MRPSO สามารถประยุกต์ใช้แก้ปัญหาการหาจุดที่ดีที่สุดในงานวิจัยด้านวิทยาศาสตร์และวิศวกรรม MRPSO สามารถประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดที่มีปริภูมิของคำตอบขนาดใหญ่มาก เป็นต้น นอกจากนี้ MRPSO ยังสามารถลดหรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า PSO ซึ่งทำให้ MRPSO ได้รับผลลัพธ์การค้นหาคำตอบที่ดีกว่า PSO ปกติ

จากผลการทดลองในบทที่ 5 MRPSO สามารถหาคำตอบได้พบทุกปัญหาที่นำมาทดลอง (ปัญหาฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ ปัญหาการเดินทางของพนักงานขาย) แต่การที่ MRPSO สามารถหาคำตอบพบได้นั้นก็ เพราะการกำหนดจำนวนของอนุภาคและรอบของการค้นหา รวมทั้งพารามิเตอร์ของการกลายพันธุ์และรีโพอิชชันนั้นเหมาะสมกับปัญหาที่แก้ไขจึงสามารถหาคำตอบพบ แต่ในการใช้งานจริงเราไม่ทราบคำตอบของปัญหาที่นำขั้นตอนวิธีไปแก้ไขจึงไม่ทราบว่าพารามิเตอร์ที่ใช้นี้สามารถแก้ปัญหาได้เหมาะสมหรือไม่ โดยปกติแล้ว PSO เองก็ต้องมีการกำหนด

พารามิเตอร์อยู่บางส่วน แต่สำหรับวิธีที่นำเสนอต้องมีพารามิเตอร์เพิ่มขึ้นจาก PSO ปกติ จึงทำให้ขั้นตอนวิธีมีความยุ่งยากในการประยุกต์ใช้งาน ดังนั้นน่าจะจะมีขั้นตอนการปรับพารามิเตอร์เหล่านี้ให้เหมาะสมกับปัญหาที่นำไปประยุกต์ใช้ ซึ่งอาจจะทำให้ประสิทธิภาพในการค้นหาคำตอบดีขึ้นและประยุกต์ใช้งานได้ง่ายขึ้น นอกจากนี้วิธีที่นำเสนอนี้ สำหรับ MXUPG ขั้นตอนการกลายพันธุ์มีการเดาคำตอบจากการกลายพันธุ์ ถ้าผลลัพธ์จากการกลายพันธุ์ที่ได้แย่กว่าเดิม คำตอบเหล่านั้นจะถูกทิ้งโดยเปล่าประโยชน์ ซึ่งเป็นการสิ้นเปลืองทรัพยากรการค้นหา และในแต่ละครั้งที่เดาคำตอบนั้นเปรียบเทียบกับว่ามีจำนวนของอนุภาคเพิ่มขึ้น ซึ่งในบางปัญหาการมีจำนวนอนุภาคมากอาจจะทำให้ได้ผลลัพธ์จากการค้นหาดีกว่าการกลายพันธุ์ ดังนั้นควรจะปรับจำนวนอนุภาคกับจำนวนการกลายพันธุ์ ให้เหมาะสมกับปัญหาที่นำไปแก้ไข และสุดท้ายสำหรับ RPG ถึงแม้จะมีขั้นตอนการรีเซ็ต PBEST และ GBEST ซึ่งทำให้อนุภาคเคลื่อนมีโอกาสเข้าสู่การค้นหาคำตอบในบริเวณใหม่ แต่เนื่องจากวิธี RPG เป็นขั้นตอนการค้นหาแบบสุ่ม จึงไม่มีอะไรรับประกันว่าการค้นหานั้นจะมีการวนกลับมาค้นหาในบริเวณเดิม ซึ่งทำให้รอบในการค้นหานั้นไม่เกิดประโยชน์ ดังนั้นน่าจะจะมีขั้นตอนที่บังคับให้ RPG ค้นหาในบริเวณอื่นนอกเหนือจากบริเวณเดิมที่เคยค้นหา ซึ่งอาจจะทำให้ประสิทธิภาพการค้นหานั้นดีขึ้น

ถึงแม้ผลการทดลองในบทที่ 5 วิธีที่นำเสนอ (MRPSO) สามารถหาคำตอบที่ดีที่สุดของปัญหาที่นำมาทดลองพบทุกปัญหา แต่ผลการทดลองในบทที่ 5 ก็ไม่ใช่ตัวแทนของปัญหาการหาจุดที่ดีที่สุดที่มีความซับซ้อนมากหรือตัวแทนของปัญหาการหาจุดที่ดีที่สุดทั้งหมดในโลกเป็นแค่ตัวแทนของปัญหาการหาจุดที่ดีที่สุดที่มีความซับซ้อนในระดับหนึ่งเท่านั้น ดังนั้น MRPSO ไม่ได้รับประกันว่าจะสามารถแก้ปัญหาการหาจุดที่ดีที่สุดได้ทุกกรณี เพราะถ้าหากนำ MRPSO ไปประยุกต์ใช้กับปัญหาการหาจุดที่ดีที่สุดที่มีความซับซ้อนมาก MRPSO ก็อาจจะเกิดปัญหาการติดในจุดสุดสัมพัทธ์และไม่สามารถหาคำตอบที่ดีที่สุดพบตัวอย่างเช่น ถ้าหากเพิ่มจำนวนมิติของฟังก์ชันมาตรฐานเป็น 100 มิติทุกฟังก์ชันแล้วกำหนดพารามิเตอร์เท่าเดิม คำตอบที่ได้จากการค้นหาด้วย MRPSO อาจจะไม่พบคำตอบที่ดีที่สุดของฟังก์ชันมาตรฐานที่นำมาทดลองนี้ หรือในกรณีปัญหาถุงเป้หลายมิติ ถ้าหากกำหนดให้มีสิ่งของที่ต้องพิจารณาเลือก 500 ชิ้น คำตอบที่ได้จากการค้นหาด้วย MRPSO อาจจะไม่พบคำตอบที่ดีที่สุดของปัญหาถุงเป้หลายมิติที่นำมาทดลองนี้ หรือในกรณีปัญหาการเดินทางของพนักงานขาย ถ้าหากกำหนดให้มีจำนวนเมือง 100 เมือง คำตอบที่ได้จากการค้นหาด้วย MRPSO อาจจะไม่พบคำตอบที่ดีที่สุดของปัญหาการเดินทางของพนักงานขายที่นำมาทดลองนี้ เป็นต้น ดังนั้น MRPSO ไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้อย่างสมบูรณ์ เพียงแต่ลดปัญหาการติดในจุดสุดสัมพัทธ์หรือแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้ถ้าหากปัญหาการหาจุดที่ดีที่สุดนั้นมีไม่ซับซ้อนมากเกินไป ดังนั้น MRPSO ควรจะต้องมีการศึกษาพัฒนาให้มีประสิทธิภาพในการแก้ปัญหาการหาจุดที่ดีที่สุดที่ซับซ้อนมากให้สามารถแก้หรือลดแก้ปัญหาการติดในจุดสุดสัมพัทธ์ให้ได้ดีขึ้นเพื่อให้ได้รับคำตอบที่ดีขึ้น

นอกจากนี้ MRPSO ใช้การทำงานของ PSO เป็นส่วนใหญ่ ซึ่งการทำงานของ PSO เป็นการค้นหาแบบสุ่ม ดังนั้นการค้นหาคำตอบในแต่ละรอบอาจจะได้คำตอบไม่เท่ากัน ในบางครั้งอาจจะพบคำตอบที่ดีที่สุดและในบางครั้งอาจจะไม่พบคำตอบที่ดีที่สุด ถึงแม้ผลการทดลองจะพบว่า MRPSO สามารถหาคำตอบที่ดีที่สุดในทุกชุดทดลองพบทุกครั้ง แต่นั่นก็ไม่ได้เป็นการรับประกันว่า MRPSO จะสามารถหาคำตอบที่ดีที่สุดพบทุกครั้งในทุกปัญหา เพียงแต่ผลการทดลองแสดงให้เห็นว่า MRPSO เป็นขั้นตอนวิธีที่มีความน่าเชื่อถือมากกว่า PSO ดังนั้น MRPSO ควรจะต้องมีการศึกษาพัฒนาให้สามารถรับประกันคำตอบที่ได้จากการค้นหาให้ได้รับคำตอบที่ดีเท่าเดิมทุกครั้ง

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] K. E. Parsopoulos and M. N. Vrahatis, "Recent Approaches to Global Optimization Problems through Particle Swarm Optimization", Natural Computing, pp. 235-306, 2002.
- [4] C. Maurice, "Particle Swarm Optimization", Wiley-ISTE, 2006.
- [5] J. H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan press, Ann Arbor, pp. 211, 1979.
- [6] D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning", Addison-Wesley, pp. 372, 1989.
- [7] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Objective function stretching to alleviate convergence to local minima", Nonlinear Analysis TMA 47, pp. 3419-3424, 2003.
- [8] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", Proc. Seventh Annual Conference on Evolutionary Programming, pp. 611-619, 1998.
- [9] R. C. Eberhart and J. Kennedy, "Swarm Intelligence", Morgan Kaufmann, San Mateo, 2001.
- [10] B. Qinghai, "Analysis of Particle Swarm Optimization Algorithm", Computer and Information Science, vol.3, No.1, pp. 180-184, 2010.
- [11] R. Chiabwot and K. Boontee, "Mutation Period Calculation for Particle Swarm Optimization", 1st International Symposium on Technology for Sustainability (ICETCIP 2011), pp. 213-216, 2011.
- [12] R. A. Krohling and L. S. Coelho, "PSO-E: Particle swarm with exponential distribution", IEEE World Congress on Computational Intelligence, Proceedings of IEEE Congress on Evolutionary Computation Congress on Evolutionary Computation, pp. 5577-5582, 2006.
- [13] L. MO and H. ZHENG, "Improved PSO Algorithm with Adaptive Inertia Weight and Mutation", World Congress on Computer Science and Information Engineering, pp. 627-625, 2009.
- [14] N. Hiquishi and H. Iba, "Particle Swarm Optimization with Gaussian Mutation", IEEE Conf. Swam Intelligence Symposium (SIS), pp. 72-79, 2003.
- [15] R. Thangaraj, M. Pant, and A. Abraham, "A new diversity guided particle swarm optimization with mutation", World Congress on Nature and Biologically Inspired Computing, pp. 294-299, 2009.
- [16] P.S. Andrews, "An investigation into mutation operators for particle swarm optimization", IEEE Congress on Evolutionary Computation, pp. 1044-1051, 2006.
- [17] C. Pinkey, D. Kusum and P. Millie, "Power Mutation Embedded Modified PSO for Global Optimization Problems", SEMCCO, pp. 139-146, 2010.

- [18] R. A. Krohling, "Gaussian particle swarm with jumps", the IEEE Congress on Evolutionary Computation, Edinburgh, pp. 1226-1231, 2005.
- [19] A. Stacey, M. Jancic and I. Grundy, "Particle swarm optimization with mutation", Proc. Congr. Evol. Comput, pp. 1425-1430, 2003.
- [20] Z. Qing, L. Changhe, L. Yong and K. Lishan, "Fast Multi-swarm Optimization with Cauchy Mutation and Crossover Operation", the 2nd international conference Advances in computation and Intelligence, pp. 344-352, 2007.
- [21] W. JIAO, G. LIU, and D. LIU, "Elite Particle Swarm Optimization with Mutation", Asia Simulation Conference 7th Intl. Conf. on Sys. Simulation and Scientific Computing, pp. 800-803, 2008.
- [22] R. Chiabwoot and K. Boontee, "Selective Crossover base on Fitness in Multi-Swarm Optimization", International Conference on Emerging Trends in Computer and Image Processing, pp. 12-15, 2011.
- [23] Z. An, X. Shi, and J. Zhang "Family Particle Swarm Optimization with Piecewise Mutation Probability", American Journal of Engineering and Technology Research, pp. 2561-2565, 2011.
- [24] D. Jiyang and Y. Jin, "Adaptive Mutation Based Particle Swarm Optimization Algorithm", UKACC International Conference on Control 2012 Cardiff, pp. 404-408, 2012.
- [25] J. Liu, J. Sun, and W.B. Xu, "Quantum-behaved particle swarm optimization with mutation operator", Proc. 17th IEEE International Conference on Tools with Artificial Intelligence, pp. 237-240, 2005.
- [26] S. Xianjun, W. Kaiping, W. Deming, T. Yala, and L. Yuanxiang, "An Dynamic Adaptive Dissipative Particle Swarm Optimization with Mutation Operation", Control and Automation, pp. 586-589, 2007.
- [27] J. Lin, X. Fan, and Z. Qu, "An improved particle swarm optimization with mutation based on similarity", Third International Conference on Natural Computation, Vol. 4, pp. 824-828, 2007.
- [28] H.R. Li and Y.L. Gao, "Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation", Second International Conference on Information and Computing Science, pp. 66-69, 2009.
- [29] H. Li and Y. Gao, "Particle Swarm Optimization Algorithm with Adaptive Threshold Mutation", International Conference on Computational Intelligence and Security, pp. 129-132, 2009.
- [30] L. Zhen-Su, H. Zhi-Rong and D. Juan, "Particle swarm optimization with adaptive mutation", Front Electr. Electron. Eng. China, pp. 99-104, 2006.
- [31] L. Haiyan, S. Pichet, H.S. Yong and D. Tharam, "Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function", International Journal of Electrical Power & Energy Systems, pp. 921-935, 2010.
- [32] R. Chiabwoot and K. Boontee, "A modified particle swarm optimization with dynamic mutation period", Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1-6, 2014.
- [33] A. ALFI, "PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems". Acta Automatica Sinica, pp. 541-549, 2011.

- [34] H. Gao and W. Xu, "Particle swarm algorithm with hybrid mutation strategy Applied Soft Computing", pp. 5129-5142, 2011.
- [35] L. Fung-Bao, "Inverse estimation of wall heat flux by using particle swarm optimization algorithm with Gaussian mutation", *International Journal of Thermal Sciences*, vol. 54, pp. 62-69, 2012.
- [36] J. Tang and X. Zhao, "Particle Swarm Optimization with Adaptive Mutation WASE", *International Conference on Information Engineering*, pp. 234-237, 2009.
- [37] P. Millie, T. Radha, V. P. Singh and A. Ajith, "Particle Swarm Optimization Using Sobol Mutation", *Int. Journal of Simulation Systems, Science and Technology*, Vol. 10, pp. 91-96, 2009.
- [38] X. Wu and M. Zhong, "Particle Swarm Optimization Based on Power Mutation", *ISECS International Colloquium on Computing, Communication, Control, and Management*, pp. 464-467, 2009.
- [39] Z. Xiaoxia, "A modified particle swarm optimization with differential evolution mutation", In *Sixth International Conference on Natural Computation*, pp. 2530-2534, 2010.
- [40] C. Li, S. Yang and I.A. Korejo, "An Adaptive Mutation operator for Particle Swarm", *Proceedings of the 2008 UK Workshop on Computational Intelligence*, pp. 165-170, 2008.
- [41] J. Tang and X. Zhao, "A hybrid particle swarm optimization with adaptive local search", *Journal of Networks*, pp. 411-418, 2010.
- [42] J. Tang and X. Zhao, "Particle swarm optimization using adaptive local search", *International Conference on Future BioMedical Information Engineering*, art., pp. 300-303, 2009.
- [43] C. C. A. Omar, R. Andrew, and L. F. Rafael, "A PSO-based algorithm with local search for multimodal optimization without constraints", *Informatica (CLEI), Conferencia Latinoamericana En*, pp. 1-7, 2012.
- [44] C. Lei, "Particle Swarm Optimization with a Novel Mutation Operator", *International Conference on Mechatronic Science, Electric Engineering and Computer*, pp. 970-973, 2011.
- [45] M. Pant, R. Thangaraj and A. Abraham, "Particle swarm optimization using adaptive mutation", *Proc. 19th International Conference on Database and Expert Systems Application*, pp. 519-523, 2008.
- [46] R. Hsin and C. Kuo-Hua, "Particle swarm optimization with mutation for the inspection allocation in reentrant production systems", *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 1025-1030, 2011.
- [47] M. S. Mohamed, G. E. R. Mona and K. A. Hanan, "A New Fixed Channel Assignment Algorithm Using Adaptive Mutation Particle Swarm Optimization", *Innovations on Communication Theory*, pp. 1-5, 2012.
- [48] L. Xuedan, W. Qiang, L. Haiyan and L. Lili, "Particle Swarm Optimization with Dynamic Inertia Weight and Mutation", *Third International Conference on Genetic and Evolutionary Computing*, pp. 620-623, 2009.
- [49] Y. Gao and Y. Duan, "A New Particle Swarm Optimization Algorithm with Adaptive Mutation Operator", *Second International Conference on Information and Computing Science*, pp. 58-61, 2009.

- [50] F. van Den Bergh, "An analysis of particle swarm optimizers", Ph.D. Dissertation, University of Pretoria, 2001.
- [51] L. Ning, S. Debao, C. Yigang, and Z. Tong, "Particle swarm optimization with mutation operator", *Computer Engineering and Applications*, pp. 12-14, 2004.
- [52] R. Chiabwoot and K. Boontee, "A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period", *Advances in Intelligent Systems and Computing*, Vol. 265, pp. 33-42, 2014.
- [53] S. C. Esquivel and C. A. Coello Coello, "On the use of particle swarm optimization with multimodal functions", the 2003 Congress on Evolutionary Computation. IEEE Press, pp. 1130–1136, 2003.
- [54] M. Marcin and S. Czesław, *Test functions for optimization needs*, 2005.
- [55] A. P. Ernesto, "Multivariate Test Functions Library in C for Unconstrained Global Optimization", Department of Mathematics U.P. Diliman, 2005.
- [56] H. Abdel-Rahman, "Global Optimization Test Problems". [http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm]
- [57] S. Sonja, B. Derek, and U. F. Simon, "Virtual Library of Simulation Experiments: Test Functions and Datasets". [<http://www.sfu.ca/~ssurjano>]
- [58] DEAP Project, "Benchmarks". [<http://deap.gel.ulaval.ca/doc/0.9/api/benchmarks.html>]
- [59] Computational Intelligence Laboratory, Peking University, "Benchmark functions for PSO". [http://www.cil.pku.edu.cn/resources/benchmark_pso]
- [60] G. Andrea, "Global Optimization Benchmarks and AMPGO". [http://infinity77.net/global_optimization]
- [61] J.E. Beasley, "ORLib – Operations Research Library". [<http://brunel.ac.uk/~mastjib/jeb/orlib/files/mknap2.txt>]
- [62] G. Reinelt, "TSPLIB - A traveling salesman problem library", *ORSA Journal on Computing*, pp. 376–384, 1991.
- [63] History of Optimization Problem and Models, "iPAS's Technical notes", [<http://ipass.wordpress.com/2009/12/17>]
- [64] Wikipedia, the free encyclopedia, "Optimization problem". [http://en.wikipedia.org/wiki/Optimization_problem]
- [65] K. Erwin, "Advanced Engineering Mathematics", 9th edition John Wiley & Sons, Inc., 1999.
- [66] F. Rothlauf, "Design of Modern Heuristics, Optimization Problems", Springer, Berlin, 2011.
- [67] R. Y. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization", *Methodology and Computing in Applied Probability*, pp. 127–190, 1999.
- [68] W. J. Stephen, "Continuous Optimization (Nonlinear and Linear Programming)", Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, USA.
- [69] G. Nicholas, "An introduction to algorithms for continuous optimization", Oxford University Computing Laboratory and Rutherford Appleton Laboratory, 2006.
- [70] P. B. Stephen and L. Vandenberghe, "Convex Optimization", Cambridge University, pp. 129, 2004.
- [71] A.E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing", Springer, Berlin, 2003.

- [72] Wikipedia, the free encyclopedia, "Local optimum". [http://en.wikipedia.org/wiki/Local_optimum]
- [73] Wikipedia, the free encyclopedia, "Global optimum". [http://en.wikipedia.org/wiki/Global_optimum]
- [74] Wikipedia, the free encyclopedia, "Knapsack problem". [http://en.wikipedia.org/wiki/Knapsack_problem]
- [75] C. Chiemchaisri, J. P. Juanga, and C. Visvanathan, "Municipal solid waste management in Thailand and disposal emission inventory", Springer Science and Business Media, pp. 34-41, 2006.
- [76] R. A. Sarker and C. S. Newton, "Optimization Modelling A Practical Approach", CRC Press, Taylor & Francis Group, 2008.
- [77] เฉียบวุฒิ รัตนวิไลสกุล, "การหลีกเลี่ยง โลคอลล ออปติ멈 สำหรับ พาร์ติเคิล สวอรัม ออปติไมเซชัน โดยการปรับรอบการกลายพันธุ์และการเปลี่ยนค่า โกลบอล เบสท์", วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2556.
- [78] D. Meera, "Traveling Salesman Problem", Wired.com sciencenews.org. [<http://www.dogonews.com/2012/9/30/scientists-learn-how-bumblebees-solve-complex-traveling-salesman-problem-so-effortlessly>]
- [79] X.S. Yang and S. Koziel, "Computational optimization, modeling and simulation – a paradigm shift", Procedia Computer Science, pp. 1291–1294, 2010.
- [80] F. Glover and M. Laguna, "Tabu Search", Kluwer Academic Publishers, USA, 1997.
- [81] A. Hertz, E. Taillard, and D. Werra, "A Tutorial on Tabu Search". [<http://www.cs.colostate.edu/~whitley/CS640/hertz92tutorial.pdf>]
- [82] Wikipedia, the free encyclopedia, "Travelling salesman problem". [http://en.wikipedia.org/wiki/Travelling_salesman_problem]
- [83] A. Michael, W. Stefan, W. Stephan and B. Andreas, "Genetic Algorithms and Genetic Programming", Modern Concepts and Practical, 2009.
- [84] R. Chiabwoot and K. Boontee, "A modified particle swarm optimization with mutation and reposition, International Journal of Innovative Computing", Information and Control, Volume 10, pp. 1-18, 2014.
- [85] V.S. Pappala, I. Erlich, K. Rohrig, and J.A. Dobschinski, "A stochastic modal for the optimal operation of a wind–thermal power system", IEEE Trans. Power Syst., pp. 940–950, 2009.
- [86] P. Acharjee and S.K. Goswami, "A decoupled power flow algorithm using particle swarm optimization technique", Energy Convers. Manage, pp. 2351–2360, 2009.
- [87] S. Dutta and S.P. Singh, "Optimal rescheduling of generators for congestion management based on particle swarm optimization", IEEE Trans. Power Syst., 2008.
- [88] S. Kiranyaz, T. Ince, A. Yildirim and M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization", Neural Networks, pp. 1448–1462, 2009.
- [89] H.L. Wei, S.A. Billings, Y.F. Zhao and L.Z. Guo, "Lattice dynamical wavelet neural networks implemented using particle swarm- optimization for spatio-temporal system identification", IEEE Trans. Neural Network, pp. 181–185, 2009.

- [90] C.J. Lin, C.H. Chen and C.T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications", *IEEE Trans. Syst. Man Cybernetics Part C*, pp. 55–68, 2009.
- [91] L. Zhao, F. Qian, Y. Yang and Y. Zeng, H. Su, "Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization", *Appl. Soft Comput.*, pp. 938–944, 2010.
- [92] C. Hanyong, O. Shunichi, Y. Kazuho and O. Hiroaki, "Improvement of Particle Swarm Optimization Application of the Mutation Concept for the Escape from Local Minima", 8th International Conference on Supply Chain Management and Information Systems, pp.1-5, 2010.
- [93] T. KRINK, J. S. VESTERSTROM and J. RIGET, "Particle swarm optimization with spatial particle extension", *Proceedings of IEEE Congress on Evolutionary Computation*. Honolulu, Hawaii USA, 2002.
- [94] M. Kong and P. Tian, "Apply the Particle Swarm Optimization to the Multidimensional Knapsack Problem", *Artificial Intelligence and Soft Computing (ICAISC 2006)*, Vol. 4029, pp. 1140-1149, 2006.
- [95] E. I. George and G. B. Mounir, "Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks", *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA*, pp. 3901 - 3908, 2009.
- [96] Z. Zhi-Hui, Z. Jun and L. Yun, C. S. Henry, "Adaptive Particle Swarm Optimization", *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, VOL. 39, NO. 6, 2009.
- [97] A. Ratnaweera, S. K. Halgamuge and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [98] D. Dumitrescu, B. Lazzerini, L.C. Jain and A. Dumitrescu, "Evolutionary Computation", CRC Press, 2000.
- [99] I. Muhammad, J. Hajira, A. Mubashir, A. Qamar, B. Waqas and A. Qamar, "Opposition based PSO and Mutation Operators", 2nd International Conference on Education Technology and Computer (ICETC), pp. 506- 508, 2010.
- [100] C. A. Floudas, P. M. Pardalos, C. Adjiman, W. R. Esposito, Z. H. Gümüs, S. T. Harding, J. L. Klepeis, C. A. Meyer and C. A. Schweiger, "Handbook of Test Problems in Local and Global Optimization", Springer, The Netherlands., pp. 484, 1999.
- [101] S. Nootyaskool and B. Kruatrachue, "Hybrid Genetic Algorithm with Baum-Welch Algorithm by using Diversity Population Technique", *International Symposium on Communications and Information Technology (ISCIT)*, pp. 15-20, 2006.
- [102] R. C. Eberhart and J. Kennedy, "A discrete binary version of the particle swarm algorithm", *Proceedings of the 1997 Conference on Systems, Man, and Cybernetics*, pp. 4104-4109, 1997.
- [103] S. Lee, S. Soak, S. Oh, W. Pedrycz and M. Jeon, "Modified binary particle swarm optimization", *Progress in Natural Science*, pp. 1161-1166, 2008. -

- [104] B. C. Jagdish and D. Kusum, "A Modified Binary Particle Swarm Optimization for Knapsack Problems", *Applied Mathematics and Computation*, pp. 11042–11061, 2012.
- [105] L. Huang, K. Wang, C. Zhou, W. Pang, L. Dong and L. Peng, "particle swarm optimization for traveling salesman problems", *Journal of Ji Ling university*, pp. 477 – 480, 2003.
- [106] L. Zhong and H. Lei, "A Mixed Discrete Particle Swarm Optimization for TSP", 3rd International Conference on Advanced Computer Theory and Engineering, pp. 208 – 211, 2010.
- [107] S. Senyu and Y. Toyoda, "An approach to linear programming with 0–1 variables", *Management Science*, pp. 196–207, 1967.
- [108] H.M. Weingartner and D.N. Ness, "Methods for the solution of the multidimensional 0/1 knapsack problem", *Operations Research*, pp. 83–103, 1967.
- [109] W. Shih, "A branch and bound method for the multiconstraint zero-one knapsack problem", *Journal of Operations Research Society*, pp. 369–378, 1979.
- [110] A. Freville and G. Plateau, "Hard 0-1 multiknapsack test problems for size reduction methods", *Investigation Operativa*, pp. 251-270, 1990.
- [111] K. Deep and J.C. Bansal, "A socio-cognitive particle swarm optimization for multi-dimensional knapsack problem", *First International Conference on Emerging Trends in Engineering and Technology (ICETET)*, pp. 355–360, 2008.
- [112] C. Maurice, "Particle Swarm Optimization", John Wiley & Sons, 2010.
- [113] J. Momin and Y. Xin-She, "A literature survey of benchmark functions for global optimization problems", *Int. Journal of Mathematical Modeling and Numerical Optimization*, Vol. 4, No. 2, pp. 150–194, 2013.
- [114] S. Martello and P. Toth, "Bin-packing problem in Knapsack Problems", *Algorithms and Computer Implementations*. Wiley, pp. 221–245, 1990.
- [115] Y. Zhou, Z. Kuang and J. Wang, "A Chaotic Neural Network Combined Heuristic Strategy for Multidimensional Knapsack Problem", *Proc. L. Kang et al. (Eds.): ISICA*, pp.715–722, 2008.
- [116] P.C. Chu and J.E. Beasley, "A Genetic Algorithm for the Multidimensional Knapsack Problem", *Journal of Heuristics*, vol. 4, pp.63–86, 1998.
- [117] J. Deane and A. Agarwal, "Neural, Genetic, And Neurogenetic Approaches For Solving The 0-1 Multidimensional Knapsack Problem", *International Journal of Management & Information Systems*, Vol. 17, pp. 43-54, 2013.
- [118] M-A. Khanesar, M. Teshnehlab and M-A. Shoorehdeli, "A Novel Binary Particle Swarm Optimization", the 15th Mediterranean Conference on Control & Automation, 2007.
- [119] Z. Jing-Ru, Z. Jun, L. Tat-Ming and L. R. Michael, "A hybrid particle swarm optimization–back-propagation algorithm for feed forward neural network training", *Applied Mathematics and Computation*, pp. 1026–1037, 2007.
- [120] X . Rui , C . W. Donald and F. L . Ronald, "Inference of Genetic Regulatory Networks with Recurrent Neural Network Models Using Particle Swarm Optimization", *IEEE/ACM transactions on computational biology and bioinformatics*, Vol. 4, pp. 4, 2007.
- [121] DW. Van der Merwe and AP. Engelbrecht, "Data Clustering using Particle Swarm Optimization", *The Congress on Evolutionary Computation*, pp. 215-220, 2003.

- [122] S. Rana, S. Jasola and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering", *Artificial Intelligence Review*, pp. 211-222, 2011.
- [123] R. Sandeep, J. Sanjay and K. Rajesh, "A review on particle swarm optimization algorithms and their applications to data clustering", *Journal Artificial Intelligence*, Vol. 35, pp. 211-222, 2011.
- [124] Z. Yu-Jun and C. Sheng-Yong, "particle swarm optimization for multiobjective transportation planning", *Applied Intelligence*, pp. 202-216, 2013.
- [125] H. ZHI-FENG , H. HAN and Y. XIAO-WEI, "A NOVEL PARTICLE SWARM OPTIMIZATION ALGORITHM FOR SOLVING TRANSPORTATION PROBLEM", the Fifth International Conference on Machine Learning and Cybernetics, pp. 13-16, 2006.
- [126] P. Suraj, W. Linlin, G. M. Siddeswara and B. Rajkumar, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", *Advanced Information Networking and Applications (AINA)*, 24th IEEE International Conference on, pp. 400 - 407, 2010.
- [127] B. N. Azlina , A. W. Mohemmed and A. Y. Mohammad, "A Wireless Sensor Network Coverage Optimization Algorithm Based on Particle Swarm Optimization and Voronoi Diagram", *Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control*, Okayama, Japan, pp. 602 - 607, 2009.
- [128] Y. Alinejad-Beromi, M. Sedighzadeh and M. Sadighi, "Particle Swarm Optimization for Siting and Sizing of Distributed Generation in Distribution Network to Improve Voltage Profile and Reduce THD and Losses", *Universities Power Engineering Conference (UPEC) 43rd International*, pp. 1 - 5, 2008.
- [129] V. d. Yamille, V. K. Ganesh, M. Salman, H. Jean-Carlos and H. G. Ronald, "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", *IEEE transactions on evolutionary computation*, Vol. 12, pp. 171-179, 2008.
- [130] M. R. AlRashidi and M. E. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Electric Power Systems", *IEEE transactions on evolutionary computation*, Vol. 13, pp. 913 - 918, 2009.
- [131] P. Jim and M. Alcherio, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization", *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, pp. 332 - 339, 2007.
- [132] H. M. James, "A Distributed Particle Swarm Optimization Algorithm for Swarm Robotic Applications", *IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada*, pp. 1678 - 1685, 2006.
- [133] Q. YUAN-QING, S. DE-BA, L. MNG and C. YI-GANG, "path planning for mobile robot using the particle swarm optimization with mutation operator", *Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai*, pp. 2473-2478, 2004.
- [134] J. Yi-Xiong, C. Hao-Zhong, Y. Jian-yong and Z. Li, "New discrete method for particle swarm optimization and its application in transmission network expansion planning", *Electric Power Systems Research*, pp. 227-233, 2007.
- [135] D.Y. Sha and H. Cheng-Yu, "A hybrid particle swarm optimization for job shop scheduling problem", *Computers & Industrial Engineering*, pp. 791-808, 2006.

- [136] P. Quan-Ke, M. F. Tasgetiren and L. Yun-Chia, "A discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem Computers & Operations Research", pp. 2807 – 2839, 2008.
- [137] L. Ching-Jong, T. Chao-Tang and L. Pin, "A discrete version of particle swarm optimization for flowshop scheduling problems", Computers & Operations Research, pp. 3099–3111, 2007.

ภาคผนวก

ภาคผนวก ก

รายละเอียดของฟังก์ชันมาตรฐาน

การค้นหาค่าตอบของฟังก์ชันมาตรฐาน คือการค้นหาค่าที่ดีที่สุดของฟังก์ชันมาตรฐาน สำหรับฟังก์ชันมาตรฐานที่รู้จักที่ดีที่สุด คือฟังก์ชันมาตรฐานชนิดที่ไม่มีเงื่อนไขบังคับ (Unconstrained Optimization) โดยฟังก์ชันจะเขียนในรูป $f(x)$ ประกอบไปด้วย x แทนด้วยมิติ ซึ่งภายในฟังก์ชัน มีหลายมิติจึงเขียนในรูป $x = \{x_1, x_2, \dots, x_n\}$ สำหรับขอบเขตของฟังก์ชันแสดงในช่วง $x_{lower} < x_i < x_{upper}$ ความซับซ้อนของฟังก์ชันมาตรฐานกำหนดได้จากจำนวนมิติ สำหรับการทดลองนี้ ใช้จำนวนมิติ 40 ถึง 100 มิติ ซึ่งมีความซับซ้อนสูงเพื่อใช้ในการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีที่ใช้ค้นหาค่าการหาจุดที่ดีที่สุด

ฟังก์ชันมาตรฐานที่นำมาทดลองได้อ้างอิงจาก [54 - 60, 100, 101, 113] โดยมีทั้งหมด 24 ฟังก์ชัน สำหรับรายละเอียดของฟังก์ชันนั้นแสดงไว้ในตาราง ก.1 และรูปฟังก์ชันได้แสดงไว้ในรูป ก.1 ถึง ก.24 เป็นลักษณะ 2 มิติเพื่อให้ง่ายต่อการมองลักษณะของฟังก์ชันและสมการของฟังก์ชันได้แสดงไว้ในสมการ ก.1 ถึง ก.24 โดยฟังก์ชันมาตรฐานสามารถแบ่งตามลักษณะได้ 2 แบบดังนี้

1. ฟังก์ชันมัลติโมดอล (Multimodal Function) คือฟังก์ชันที่มีจุดสุดสัมพัทธ์ โดยลักษณะจะมีความลาดเอียงหลายรูปแบบผสมผสานกัน เนื่องจากสร้างมาจากการรวมฟังก์ชันยูนิโมดอลหลายรูปแบบ ทำให้ฟังก์ชันมัลติโมดอลนี้มีความซับซ้อนมากกว่าฟังก์ชันยูนิโมดอล
2. ฟังก์ชันยูนิโมดอล (Single-modal Function) คือฟังก์ชันที่ไม่มีจุดสุดสัมพัทธ์ โดยลักษณะจะมีความลาดเอียงเพียงรูปแบบเดียว เช่นฟังก์ชันมาตรฐานที่สร้างมาจากสัญญาณ SINE เพียงอย่างเดียว ฟังก์ชันยูนิโมดอลนี้ดูเหมือนง่ายต่อการค้นหาค่าตอบแต่ในความจริงฟังก์ชันยูนิโมดอลมีส่วนประกอบของพื้นที่ราบเรียบจึงทำให้การค้นหานั้นจะเคลื่อนที่เข้าหาค่าตอบกระทำได้อย่าง

ตารางที่ ก.1 สรุปสมบัติของฟังก์ชันคณิตศาสตร์ที่ใช้ในการทดลอง

ชื่อฟังก์ชัน	ขอบเขต	จุดที่ดีที่สุด	จำนวนมิติ	คุณสมบัติ	V_{\max}, V_{\min}	$Iter_{\max}$
ACKLEY	$x \in [-32.768, 32.768]^n$	$f(x) = 0$	40	Multimodal	± 32.768	20000
GRIEWANK	$x \in [-300, 300]^n$	$f(x) = 0$	40	Multimodal	± 300	20000
RASTRIGIN	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	40	Multimodal	± 5.12	20000
ROSENBROCK	$x \in [-2.048, 2.048]^n$	$f(x) = 0$	40	Multimodal	± 2.048	20000
* SCHWEFEL	$x \in [-500, 500]^n$	$f(x) = 0$	40	Multimodal	± 500	20000
COSINE MIXTURE	$x \in [-1, 1]^n$	$f(x) = 0$	40	Multimodal	± 1	20000
EXPONENTIAL	$x \in [-1, 1]^n$	$f(x) = 0$	40	Multimodal	± 1	20000
* LEVY	$x \in [-10, 10]^n$	$f(x) = 0$	40	Multimodal	± 10	20000
* MICHALEWICZ	$x \in [0, 3.14]^n$	$f(x) = 0$	10	Multimodal	± 3.14	50000
STEP	$x \in [-100, 100]^n$	$f(x) = 0$	100	Multimodal	± 100	2000
SCHAFFER	$x \in [-100, 100]^n$	$f(x) = 0$	2	Multimodal	± 100	2000
HOLDER	$x \in [-10, 10]^n$	$f(x) = 0$	2	Multimodal	± 10	2000
BEALE	$x \in [-4.5, 4.5]^n$	$f(x) = 0$	2	Multimodal	± 4.5	2000
SHUBERT	$x \in [-10, 10]^n$	$f(x) = 0$	2	Multimodal	± 10	2000
GOLDSTEIN-PRICE	$x \in [-2, 2]^n$	$f(x) = 0$	2	Multimodal	± 2	2000
TRID	$x \in [-100, 100]^n$	$f(x) = 0$	10	Multimodal	± 100	50000
SPHERE	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	40	Unimodal	± 5.12	20000
PARALLEL HYPER-ELLIPSOID	$x \in [-5.12, 5.12]^n$	$f(x) = 0$	40	Unimodal	± 5.12	20000
ROTATED HYPER-ELLIPSOID	$x \in [-65.536, 65.536]^n$	$f(x) = 0$	40	Unimodal	± 65.536	20000
CIGAR	$x \in [-10, 10]^n$	$f(x) = 0$	40	Unimodal	± 10	20000
BROWN	$x \in [-1, 4]^n$	$f(x) = 0$	40	Unimodal	± 1	20000
MULTIMOD	$x \in [-10, 10]^n$	$f(x) = 0$	40	Unimodal	± 10	50000
ZAKHAROV	$x \in [-5, 10]^n$	$f(x) = 0$	40	Unimodal	± 5.12	50000
EASOM	$x \in [-100, 100]^n$	$f(x) = 0$	2	Unimodal	± 100	2000

หมายเหตุ สำหรับฟังก์ชันที่มีเครื่องหมาย * อยู่ข้างหน้าจะมีการกำหนดพารามิเตอร์หรือคุณสมบัติพิเศษเพิ่มเติม โดยรายละเอียดจะถูกเขียนไว้ในหัวข้อของฟังก์ชันนั้นที่จะถูกกล่าวดังต่อไปนี้

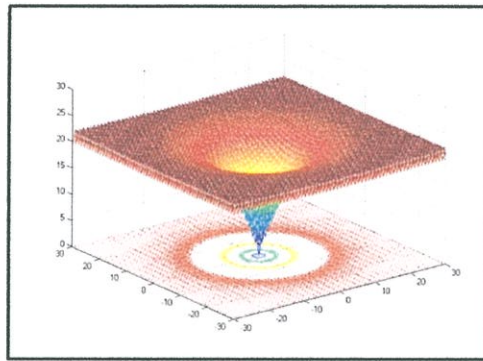
● ฟังก์ชัน ACKLEY

ฟังก์ชัน ACKLEY คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันนี้มีพื้นผิวที่ครอบคลุมด้วยการกระจายตัวแบบ EXPONENTIAL ซึ่งการกระจายตัวแบบนี้จะทำให้เกิดจุดสุดสัมพัทธ์กระจายตัวในรูปแบบปกติอย่างสม่ำเสมอบนพื้นที่ผิวของปริภูมิของคำตอบ โดยรูปของฟังก์ชันคือรูป ก.1 และสมการของฟังก์ชันคือสมการ ก.1

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

มีขอบเขตอยู่ในช่วง $x \in [-30, 30]^n$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0)$, $f(x^*) = 0$



รูปที่ ก.1 ฟังก์ชัน ACKLEY

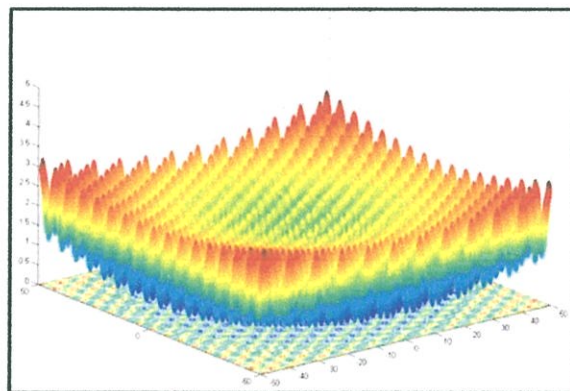
- ฟังก์ชัน GRIEWANK

ฟังก์ชัน GRIEWANK คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยมีลักษณะเหมือนกัน ฟังก์ชัน RASTRIGIN แต่มีการกระจายตัวของจุดสุดสัมพัทธ์เป็นบริเวณกว้างกว่าฟังก์ชัน RASTRIGIN อย่างไรก็ตามจุดสุดสัมพัทธ์ก็มีการกระจายตัวอย่างปกติ และสม่ำเสมอบนพื้นที่ผิวของฟังก์ชัน โดยรูปของฟังก์ชันคือรูป ก.2 และสมการของฟังก์ชันคือสมการ ก.2

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

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.2 ฟังก์ชัน GRIEWANK

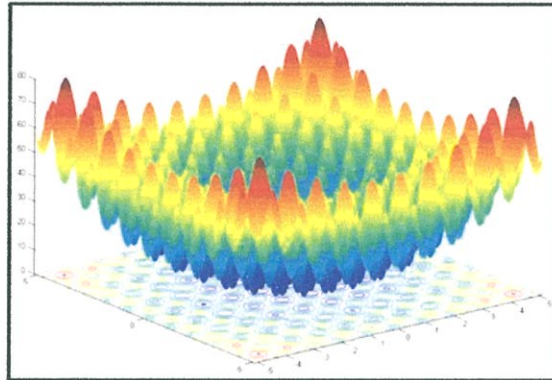
- ฟังก์ชัน RASTRIGIN

ฟังก์ชัน RASTRIGIN คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยจุดสุดสัมพัทธ์นั้นได้มีการกระจายตัวแบบ COSINE บนพื้นที่ผิวของฟังก์ชัน แต่จุดสุดสัมพัทธ์มีการกระจายตัวอย่างปกติและสม่ำเสมอบนพื้นที่ผิวของฟังก์ชัน โดยรูปของฟังก์ชันคือรูป ก.3 และสมการของฟังก์ชันคือสมการ ก.3

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

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.3 ฟังก์ชัน RASTRIGIN

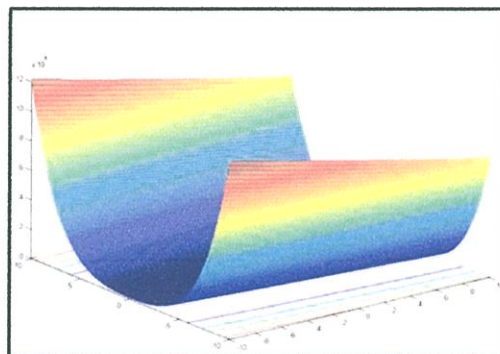
● ฟังก์ชัน ROSENBROCK

ฟังก์ชัน ROSENBROCK คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยภายในฟังก์ชันมีช่วงที่ยาวและลึกเป็นรูปแบบ HYPERBOLIC ซึ่งทำให้เกิดรูปแบบเป็นร่องหุบเขาที่กว้างยาวและราบเรียบ ดังนั้นจึงเป็นเรื่องยากที่จะลู่เข้าหาคำตอบภายในร่องหุบเขานี้ สำหรับฟังก์ชันนี้ถ้ามีจำนวนมิติมากกว่า 3 มิติจะมีสมบัติเป็นฟังก์ชันมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.4 และสมการของฟังก์ชันคือสมการ ก.4

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

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (1, \dots, 1), f(x^*) = 0$



รูปที่ ก.4 ฟังก์ชัน ROSENBROCK

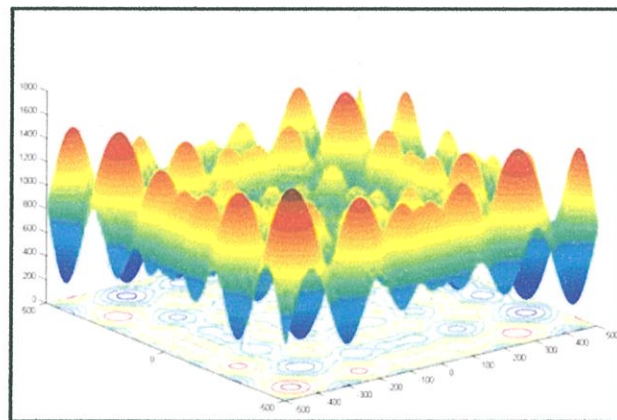
- ฟังก์ชัน SCHWEFEL

ฟังก์ชัน SCHWEFEL คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันจะยกกำลัง $1/2$ ซึ่งทำให้ยากที่จะแบ่งพื้นผิวออกเป็นส่วนๆ นอกจากนี้ในแต่ละแกนจะมีการกระจายตัวของจุดสุดสัมพัทธ์เป็นจำนวนมาก ดังนั้นการค้นหาคำตอบจึงเป็นเรื่องยากและทำได้ช้า เพราะว่ายากที่จะแบ่งพื้นผิวออกเป็นส่วนๆ โดยรูปของฟังก์ชันคือรูป ก.5 และสมการของฟังก์ชัน คือสมการ ก.5 สำหรับฟังก์ชันนี้ในการทดลองในบทที่ 5 สมการการกลายพันธุ์ที่ 3.15 และ 3.16 ได้มีการนำค่าคงที่ คือ 2.4 คูณกับ rand()

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

มีขอบเขตอยู่ในช่วง $x \in [-500, 500]^n$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (-420.96, \dots, -420.96), f(x^*) = 0$



รูปที่ ก.5 ฟังก์ชัน SCHWEFEL

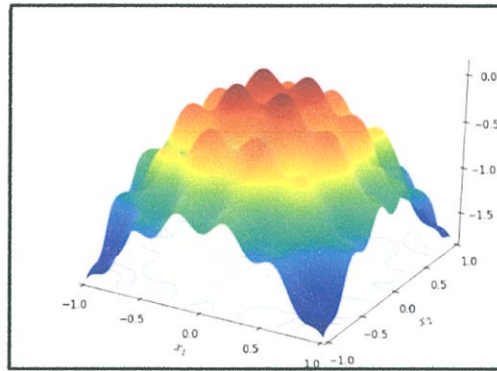
- ฟังก์ชัน COSINE MIXTURE

ฟังก์ชัน COSINE MIXTURE คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ภายในขอบเขตการค้นหา โดยรูปของฟังก์ชันคือรูป ก.6 และสมการของฟังก์ชันคือสมการ ก.6

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

มีขอบเขตอยู่ในช่วง $x \in [-1, 1]^n$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.6 ฟังก์ชัน COSINE MIXTURE

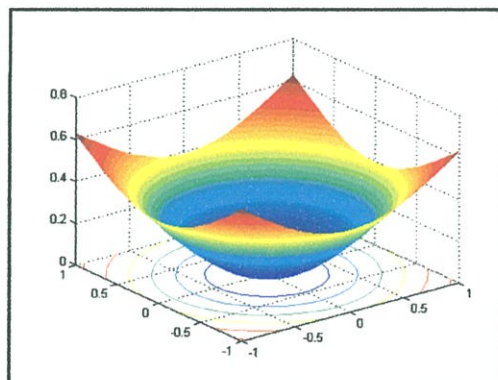
- ฟังก์ชัน EXPONENTIAL

ฟังก์ชัน EXPONENTIAL คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ในขอบเขตการค้นหา โดยรูปของฟังก์ชันคือรูป ก.7 และสมการของฟังก์ชันคือสมการ ก.7

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

มีขอบเขตอยู่ในช่วง $x \in [-1, 1]^n$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.7 ฟังก์ชัน EXPONENTIAL

- ฟังก์ชัน LEVY

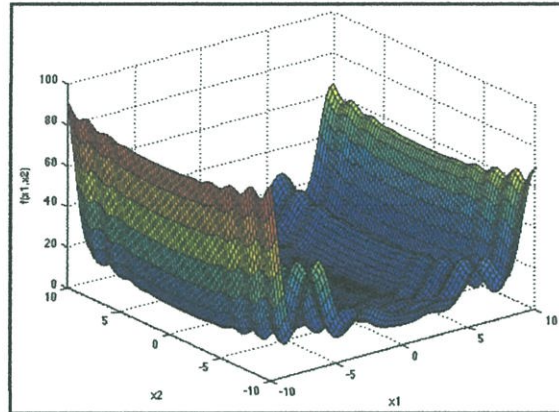
ฟังก์ชัน LEVY คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ในขอบเขตการค้นหา โดยรูปของฟังก์ชันคือรูป ก.8 และสมการของฟังก์ชันคือสมการ ก.8 สำหรับฟังก์ชันนี้ได้ลบค่าด้วย 10^{-30} เพราะถ้าแทนค่า 1 ลงในสมการนี้แล้ว คำตอบที่ได้ประมาณ 10^{-30} ดังนั้นถ้าหากลบด้วย จะได้ค่าเท่ากับ 0

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

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

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (1, \dots, 1), f(x^*) = 0$



รูปที่ ก.8 ฟังก์ชัน LEVY

● ฟังก์ชัน MICHALEWICZ

ฟังก์ชัน MICHALEWICZ คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ในขอบเขตการค้นหาสำหรับพารามิเตอร์ m เป็นตัวกำหนดความสูงชันของหุบเขาและสันเขา โดย m ยิ่งมีค่ามากก็จะยิ่งทำให้การค้นหามีความยุ่งยากมากขึ้นซึ่งโดยปกติแล้วค่า m ถูกกำหนดให้มีค่าเท่ากับ 10 โดยรูปของฟังก์ชันคือรูป ก.9 และสมการของฟังก์ชันคือสมการ ก.9 สำหรับฟังก์ชันนี้ในการทดลองในบทที่ 5 ได้มีการกำหนดจำนวนมิติเท่ากับ 10 มิติ และมีการบวกราคา 9.66015 เข้าไปเพื่อให้ค่า $f(x)$ มีค่าเท่ากับศูนย์ และได้กำหนดจำนวนประชากรเท่ากับ 1000 สำหรับกรณีที่ไม่มีการกลายพันธุ์แบบ CM สำหรับการกลายพันธุ์แบบ CM จะกำหนดจำนวนประชากรเท่ากับ 500 และกำหนดจำนวนรอบการกลายพันธุ์เท่ากับ 1

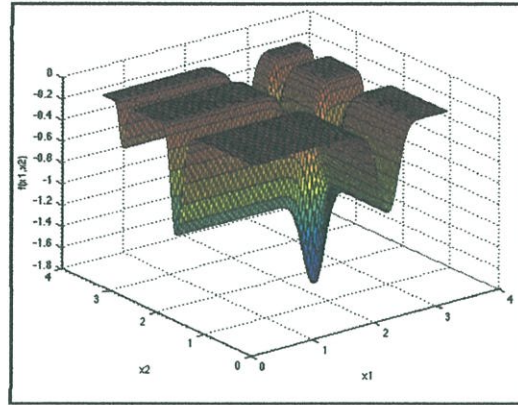
$$f(x) = -\sum_{i=1}^d \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad (ก.9)$$

มีขอบเขตอยู่ในช่วง $x \in [0, \pi]^n$

ค่าสูงสุดสัมบูรณ์ 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

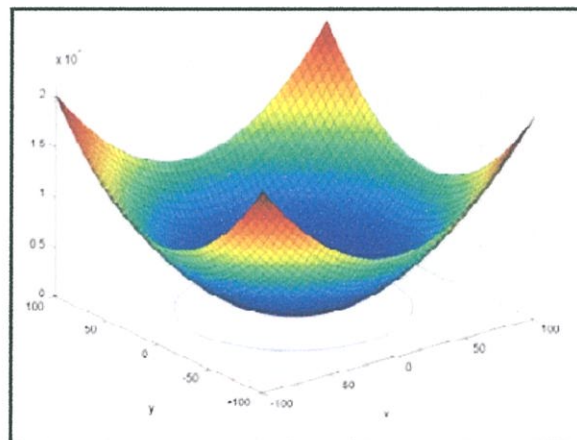
- ฟังก์ชัน STEP

ฟังก์ชัน STEP คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยลักษณะของฟังก์ชันมีทั้งค่าที่มีความชันสูงและพื้นราบลาดถึงจุดสุดสัมบูรณ์ นอกจากนี้มีการปิดค่าทศนิยมขึ้นลงทำให้คำตอบแบ่งออกเป็นชั้นๆ ส่งผลให้สมบัติเป็นฟังก์ชันมัลติโมดอล โดยรูปของฟังก์ชันคือรูป ก.10 และสมการของฟังก์ชันคือสมการ ก.10

$$f(x) = \sum_{i=0}^{n-1} (\lfloor x_i \rfloor + 0.5)^2 \quad (\text{ก.10})$$

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0.5, \dots, 0.5), f(x^*) = 0$



รูปที่ ก.10 ฟังก์ชัน STEP

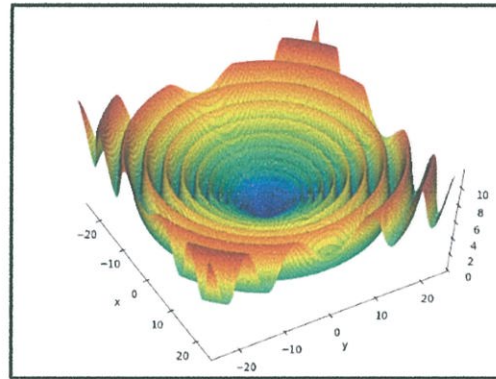
- ฟังก์ชัน SCHAFFER

ฟังก์ชัน SCHAFFER คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยมีจำนวนมิติ 2 มิติ และฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ในขอบเขตการค้นหา โดยรูปของฟังก์ชันคือรูป ก.11 และสมการของฟังก์ชันคือสมการ ก.11

$$f(x) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1.0 + 0.001 \times (x^2 + y^2))^2} \quad (\text{ก.11})$$

มีขอบเขตอยู่ในช่วง $x \in [-100, 100]$ "

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.11 ฟังก์ชัน SCHAFFER

- ฟังก์ชัน HOLDER

ฟังก์ชัน HOLDER คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล ฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ในขอบเขตการค้นหาและมีจำนวนมิติ 2 มิติ มีจำนวนจุดสุดสัมบูรณ์ 4 จุด โดยรูปของฟังก์ชันคือรูป ก.12 และสมการของฟังก์ชันคือสมการ ก.12

$$f(x) = -\left| \sin(x_1) \cos(x_2) e^{\left(1 - \sqrt{\frac{x_1^2 + x_2^2}{\pi}}\right)} \right| \quad (\text{ก.12})$$

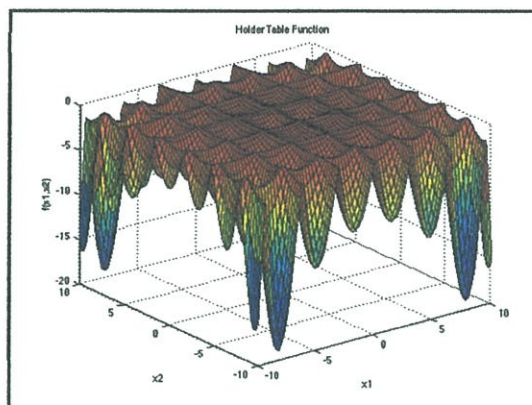
มีขอบเขตอยู่ในช่วง $x \in [-10, 10]$ "

ค่าสูงสุดสัมบูรณ์ 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.12 ฟังก์ชัน HOLDER

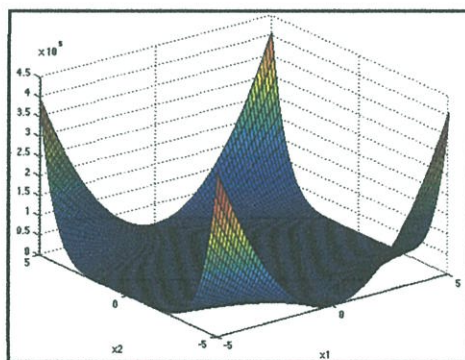
- ฟังก์ชัน BEALE

ฟังก์ชัน BEALE คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล มีจำนวนมิติ 2 มิติ สำหรับฟังก์ชันนี้มียอดแหลมอยู่ที่มุมของขอบเขตปริภูมิของคำตอบ โดยรูปของฟังก์ชันคือรูป ก.13 และสมการของฟังก์ชันคือสมการ ก.13

$$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \quad (\text{ก.13})$$

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (3, 0.5), f(x^*) = 0$



รูปที่ ก.13 ฟังก์ชัน BEALE

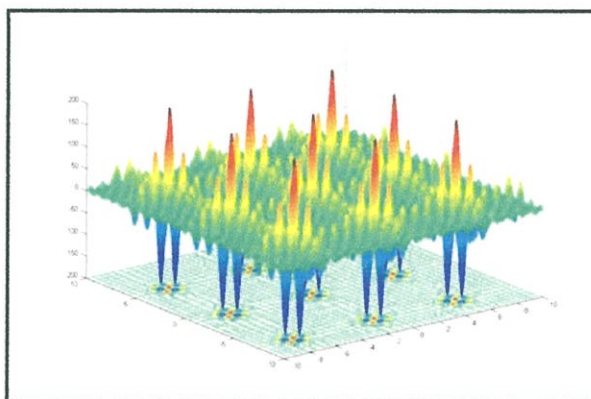
- ฟังก์ชัน SHUBERT

ฟังก์ชัน SHUBERT คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล มีจำนวนมิติ 2 มิติ และมีจุดสุดสัมบูรณ์อยู่ 18 จุดกระจายอยู่บนปริภูมิของคำตอบ โดยรูปของฟังก์ชันคือรูป ก.14 และสมการของฟังก์ชันคือสมการ ก.14 สำหรับฟังก์ชันนี้ในการทดลองในบทที่ 5 มีการบวกค่า 186.7309 เข้าไปเพื่อให้ค่า $f(x)$ มีค่าเท่ากับศูนย์

$$f(x) = \sum_{i=1}^5 i \cos((i+1)x_1 + i) \left(\sum_{i=1}^5 i \cos((i+1)x_2 + i) \right) \quad (\text{ก.14})$$

มีขอบเขตอยู่ในช่วง $x \in [-10, 10]$

ค่าสูงสุดสัมบูรณ์ คือ $f(x^*) = -186.7309$



รูปที่ ก.14 ฟังก์ชัน SHUBERT

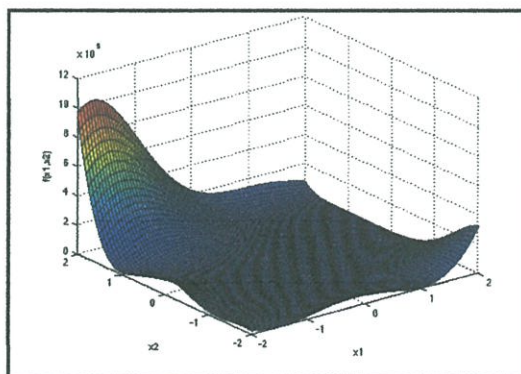
- ฟังก์ชัน GOLDSTEIN-PRICE

ฟังก์ชัน GOLDSTEIN-PRICE คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ภายในขอบเขตการค้นหาและมีจำนวนมิติ 2 มิติ โดยรูปของฟังก์ชันคือรูป ก.15 และสมการของฟังก์ชันคือสมการ ก.15 สำหรับฟังก์ชันนี้ในการทดลองในบทที่ 5 มีการลบค่า 3 เข้าไปเพื่อให้ค่า $f(x)$ มีค่าเท่ากับศูนย์

$$f(x^*) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \quad (\text{ก.15})$$

มีขอบเขตอยู่ในช่วง $x \in [-2, 2]$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, 1), f(x^*) = 3$



รูปที่ ก.15 ฟังก์ชัน GOLDSTEIN-PRICE

- ฟังก์ชัน TRID

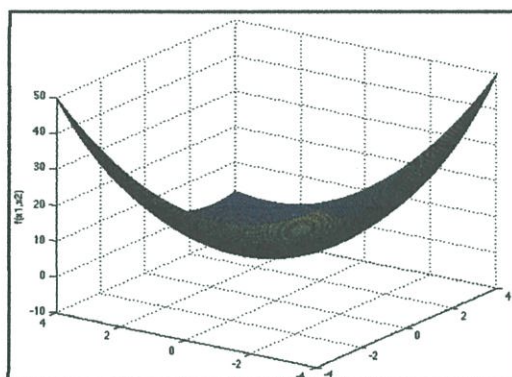
ฟังก์ชัน TRID คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันมัลติโมดอล โดยฟังก์ชันมีจุดสุดสัมพัทธ์อยู่ภายในขอบเขตการค้นหา โดยรูปของฟังก์ชันคือรูป ก.16 และสมการของฟังก์ชันคือสมการ ก.16 สำหรับฟังก์ชันนี้ในการทดลองในบทที่ 5 มีการบวกค่า 200 เข้าไปเพื่อให้ค่า $f(x)$ มีค่าเท่ากับศูนย์

$$f(x) = \sum_{i=1}^d (x_i - 1)^2 - \sum_{i=2}^d x_i x_{i-1} \quad (\text{ก.16})$$

มีขอบเขตอยู่ในช่วง $x \in [-d^2, d^2]^n$

ค่าสูงสุดสัมบูรณ์ คือ $d = 6: f(x^*) = 3$

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



รูปที่ ก.16 ฟังก์ชัน TRID

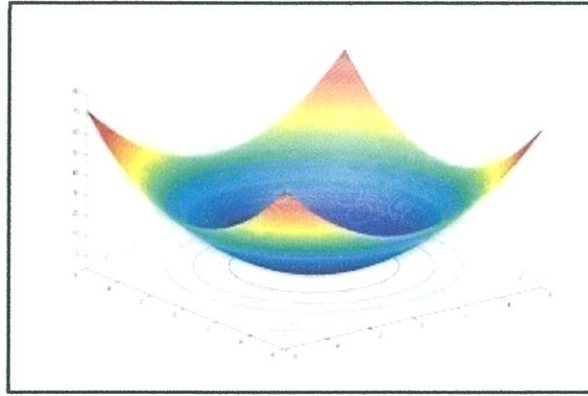
- ฟังก์ชัน SPHERE

ฟังก์ชัน SPHERE คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะเป็นฟังก์ชันต่อเนื่องที่มีโครงสร้างเป็นรูปทรงกรวยคว่ำและไม่มีจุดสุดสัมพัทธ์ มีจุดสุดสัมบูรณ์เพียงจุดเดียว โดยรูปของฟังก์ชันคือรูป ก.17 และสมการของฟังก์ชันคือสมการ ก.17

$$f(x) = \sum_{i=1}^n x_i^2 \quad (\text{ก.17})$$

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.17 ฟังก์ชัน SPHERE

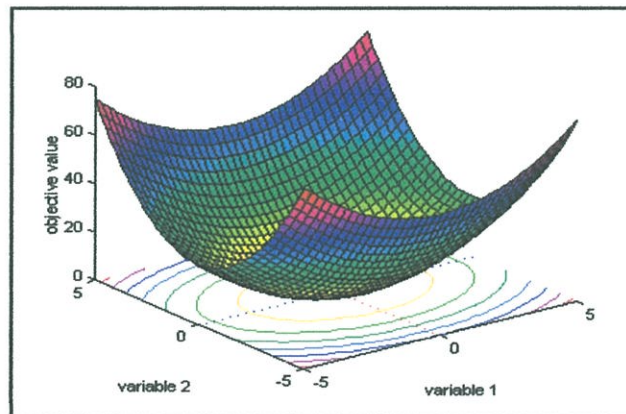
- ฟังก์ชัน PARALLEL ELLIPSOID

ฟังก์ชัน PARALLEL ELLIPSOID คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะฟังก์ชันจะมีอัตราการลาดชันเป็นลักษณะทวีคูณ เนื่องจากผลรวมของตัวแปรต้นยกกำลังสองและคูณด้วยจำนวนของมิติรวมกัน ส่งผลให้การลู่เข้าไปหาจุดสุดสัมบูรณ์ต้องใช้เวลาานาน โดยรูปของฟังก์ชันคือรูป ก.18 และสมการของฟังก์ชันคือสมการ ก.18

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

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.18 ฟังก์ชัน PARALLEL ELLIPSOID

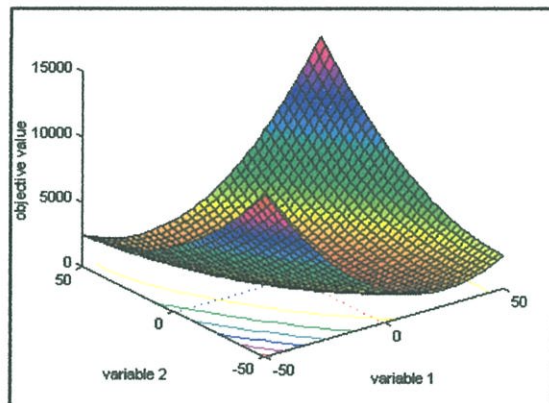
- ฟังก์ชัน ROTATED ELLIPSOID

ฟังก์ชัน ROTATED ELLIPSOID คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะฟังก์ชันจะมีอัตราการลาดชันเป็นทวีคูณเป็นผลมาจากผลรวมตัวแปรต้นสองชุดทำให้การลู่เข้าหาจุดสุดสัมบูรณ์ต้องใช้เวลาานาน โดยรูปของฟังก์ชันคือรูป ก.19 และสมการของฟังก์ชันคือสมการ ก.19

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

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.19 ฟังก์ชัน ROTATED ELLIPSOID

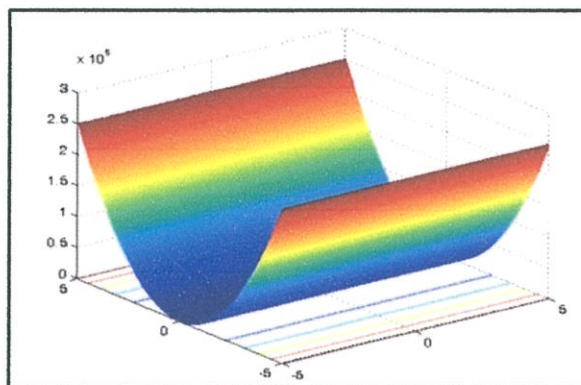
- ฟังก์ชัน CIGAR

ฟังก์ชัน CIGAR คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะของฟังก์ชันเป็นฟังก์ชันไม่มีจุดสุดสัมพัทธ์และมีจุดสุดสัมบูรณ์เพียงจุดเดียว โดยรูปของฟังก์ชันคือรูป ก.20 และสมการของฟังก์ชันคือสมการ ก.20

$$f(x) = x_0^2 + 10^6 \sum_{i=1}^N x_i^2 \quad (\text{ก.20})$$

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.20 ฟังก์ชัน CIGAR

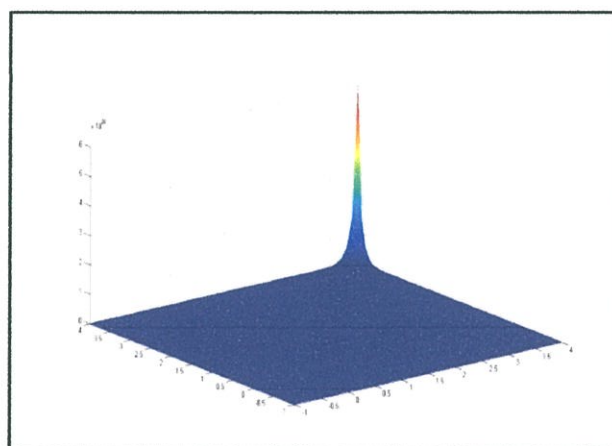
- ฟังก์ชัน BROWN

ฟังก์ชัน BROWN คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะของฟังก์ชันเป็นฟังก์ชันไม่มีจุดสุดสัมพัทธ์และมีจุดสุดสัมบูรณ์เพียงจุดเดียว โดยรูปของฟังก์ชันคือรูป ก.21 และสมการของฟังก์ชันคือสมการ ก.21

$$f(x) = \sum_{i=0}^{n-1} [(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}] \quad (\text{ก.21})$$

มีขอบเขตอยู่ในช่วง $x \in [-1,4]^n$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.21 ฟังก์ชัน BROWN

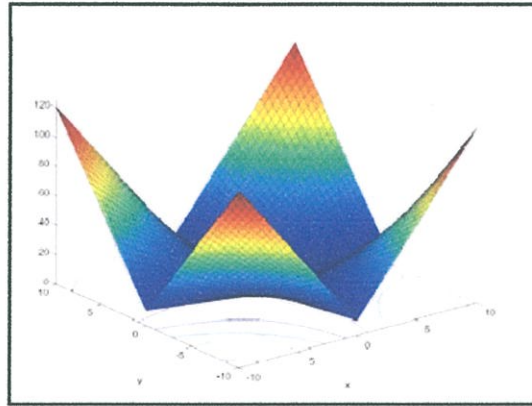
- ฟังก์ชัน MULTIMOD

ฟังก์ชัน MULTIMOD คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะของฟังก์ชันมีทั้งค่าที่มีความชันสูงและพื้นราบลาดถึงจุดสุดสัมบูรณ์ ทำให้การลู่อเข้าหาจุดสุดสัมบูรณ์ต้องใช้เวลาาน โดยรูปของฟังก์ชันคือรูป ก.22 และสมการของฟังก์ชันคือสมการ ก.22

$$f(x) = \sum_{i=1}^n |x_i| \times \prod_{i=1}^n |x_i| \quad (\text{ก.22})$$

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.22 ฟังก์ชัน MULTIMOD

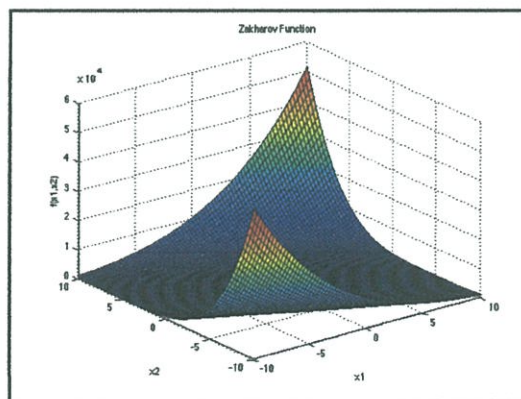
- ฟังก์ชัน ZAKHAROV

ฟังก์ชัน ZAKHAROV คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะของฟังก์ชันเป็นฟังก์ชันไม่มีจุดสุดสัมพัทธ์และมีจุดสุดสัมบูรณ์เพียงจุดเดียว โดยรูปของฟังก์ชันคือรูป ก.23 และสมการของฟังก์ชันคือสมการ ก.23

$$f(x) = \sum_{i=1}^n x_i^2 + \left[\sum_{i=1}^n \frac{i}{2} x_i \right]^2 + \left[\sum_{i=1}^n \frac{i}{2} x_i \right]^4 \quad (\text{ก.23})$$

มีขอบเขตอยู่ในช่วง $x \in [-5, 10]^n$

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (0, \dots, 0), f(x^*) = 0$



รูปที่ ก.23 ฟังก์ชัน ZAKHAROV

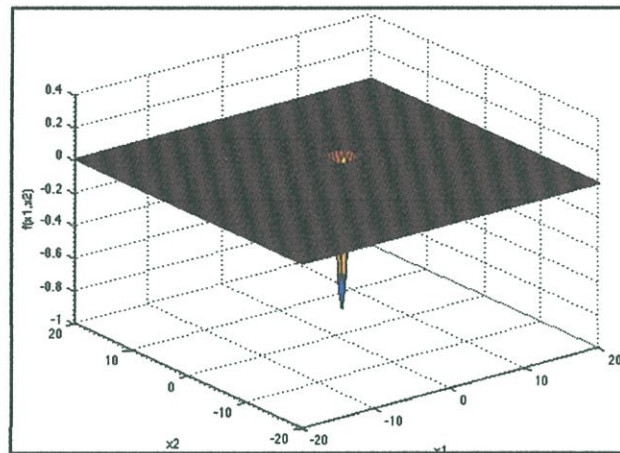
- ฟังก์ชัน EASOM

ฟังก์ชัน EASOM คือฟังก์ชันที่จัดอยู่ในกลุ่มฟังก์ชันยูนิโมดอล โดยลักษณะของฟังก์ชันคือจุดสุดสัมบูรณ์มีพื้นที่ขนาดเล็กมากเมื่อเทียบกับพื้นที่ค้นหาทั้งหมดโดยรูปของฟังก์ชันคือรูป ก.24 และสมการของฟังก์ชันคือสมการ ก.24 สำหรับฟังก์ชันนี้ในการทดลองในบทที่ 5 มีการบวกค่า 1 เข้าไปเพื่อให้ค่า $f(x)$ มีค่าเท่ากับศูนย์

$$f(x) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) \quad (\text{ก.24})$$

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

ค่าสูงสุดสัมบูรณ์ คือ $x^* = (\pi, \pi), f(x^*) = -1$



รูปที่ ก.24 ฟังก์ชัน EASOM

ภาคผนวก ข

การประยุกต์ PSO กับปัญหาถุงเป้แบบหลายมิติ

ข.1 ปัญหาถุงเป้แบบหลายมิติ

ปัญหาถุงเป้แบบหลายมิติ (The Multidimensional Knapsack Problem หรือ MKP) เป็นปัญหาประเภทเชิงการตัดสินใจ (Discrete Optimization Problem) [114] และเป็นปัญหาที่จัดอยู่ในกลุ่มปัญหาประเภทเอ็นพี-ฮาร์ด (NP-hard Problems) [114-116] ซึ่งหลักการของปัญหาถุงเป้แบบหลายมิตินี้สามารถนำไปประยุกต์ใช้กับปัญหาเชิงธุรกิจได้มากมายเช่น ปัญหาการเลือกโครงการ (Project Selection Problem) ปัญหาจัดสรรเงินลงทุน (Capital Budgeting Problem) ปัญหาจัดการคลังสินค้า (Cutting Stock Problem) ปัญหาจัดการโหลดสินค้า (Cargo Loading Problem) ปัญหาการวางแผนการผลิต (Production Planning Problem) ปัญหาการจัดตารางเวลา (Scheduling Problem) ปัญหาจัดการทรัพยากรการประมวลผล (Distributed Processor Problem) ปัญหาการจัดสรรฐานข้อมูล (Database Allocation Problem) [116, 117] ดังนั้นขั้นตอนวิธีใดที่สามารถแก้ไขปัญหาถุงเป้แบบหลายมิติได้อย่างมีประสิทธิภาพ ขั้นตอนวิธีนั้นก็จะสามารถแก้ปัญหาค่าเหมาะที่สุดที่ใกล้เคียงได้เป็นอย่างดีมีประสิทธิภาพเช่นกัน [117]

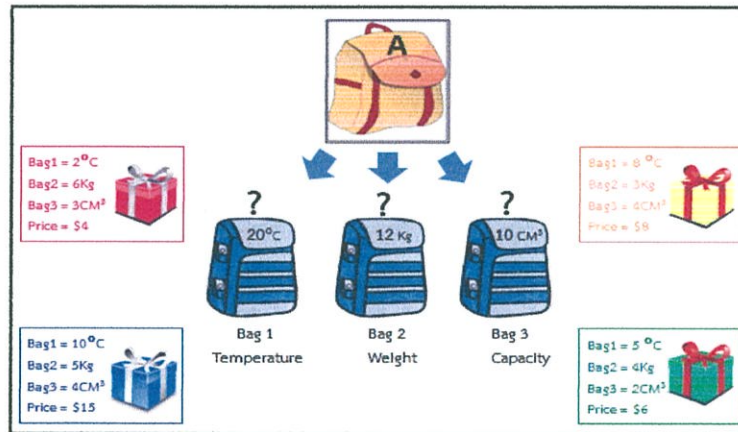
ปัญหาถุงเป้แบบหลายมิติประกอบไปด้วยถุง (Knapsacks) จำนวน m ถุง โดยแต่ละถุงมีความจุมากที่สุด (The Maximum Capacity) เท่ากับ C_j ($j=1\dots m$) ซึ่งแต่ละถุงเป็นตัวแทนของเงื่อนไขของปัญหาคืออย่างเช่น น้ำหนักที่ถุงเป้สามารถรับน้ำหนักได้ อุณหภูมิที่ถุงเป้สามารถรับน้ำหนักได้ ปริมาตรที่ถุงเป้สามารถรับได้ เป็นต้น สำหรับวัตถุทั้งหมดนั้นจะเป็นเซตของ n วัตถุ โดยแต่ละวัตถุ i ($i=1\dots n$) มีราคา (P_i) และน้ำหนัก (w_{ij}) โดยวัตถุ 1 ชิ้นจะมีน้ำหนักหรือเงื่อนไขเท่ากับจำนวนถุงเป้ตัวอย่างเช่น วัตถุชิ้นหนึ่งมีน้ำหนักของตัวเอง มีอุณหภูมิของตัวเอง มีปริมาตรของตัวเอง เป็นต้น ถ้าหากเลือกวัตถุนี้ถุงเป้จะต้องสามารถรองรับเงื่อนไขทั้งหมดของวัตถุชิ้นนี้

สำหรับจุดประสงค์ของปัญหาถุงเป้แบบหลายมิติ คือการเลือกวัตถุให้ได้ผลรวมราคามากที่สุดแต่ความจุของถุงเป้ทั้งหมดต้องสามารถรองรับน้ำหนักของวัตถุที่ใส่ทั้งหมดได้ดังรูป ข.1 ต้องการเลือกวัตถุให้มีมูลค่ารวมกันแล้วได้มากที่สุด โดยที่ถุงเป้ต้องไม่ขาด มีถุงเป้ A ซึ่งมีเงื่อนไข 3 เงื่อนไขคือ อุณหภูมิ น้ำหนัก ปริมาตร ซึ่งเปรียบเสมือนมีถุงเป้ 3 ถุงและแต่ละใบจะมีความจุไม่เท่ากัน โดยถุงเป้ใบแรกแทนอุณหภูมิที่ถุงเป้ A จะรองรับ ได้ มีความจุ 20 องศาเซลเซียส ถุงเป้ใบที่สองแทนน้ำหนักที่ถุงเป้ A จะรองรับ ได้ มีความจุ 12 กิโลกรัม และถุงเป้ใบที่สามแทนปริมาตรที่ถุงเป้ A จะรองรับ ได้ มีความจุ 15 ลูกบาศก์เซนติเมตร โดยมีวัตถุให้เลือก 4 ชิ้นแต่ละชิ้นไม่สามารถแบ่งแยกชิ้นส่วน ถ้าเลือกวัตถุชิ้นแรกจะทำให้มีอุณหภูมิเพิ่มขึ้น 2 องศาเซลเซียสสำหรับถุงเป้ใบแรก มีน้ำหนักเพิ่มขึ้น 6 กิโลกรัมสำหรับถุงเป้ใบที่สอง มีปริมาตรเพิ่มขึ้น 3 ลูกบาศก์เซนติเมตรสำหรับถุงเป้ใบที่สาม ซึ่งวัตถุนี้มีมูลค่า 4 บาท ถ้าเลือกวัตถุชิ้นที่สองจะทำให้มีอุณหภูมิเพิ่มขึ้น 8 องศาเซลเซียสสำหรับถุงเป้ใบแรก มีน้ำหนักเพิ่มขึ้น 3 กิโลกรัมสำหรับถุงเป้ใบที่สอง มีปริมาตรเพิ่มขึ้น 4 ลูกบาศก์เซนติเมตรสำหรับถุงเป้ใบที่สาม ซึ่งวัตถุนี้มีมูลค่า 8 บาท ถ้าเลือกวัตถุชิ้นที่สามจะทำให้มีอุณหภูมิเพิ่มขึ้น 10 องศาเซลเซียสสำหรับถุงเป้ใบแรก มีน้ำหนักเพิ่มขึ้น 5 กิโลกรัมสำหรับถุงเป้ใบที่สอง มีปริมาตรเพิ่มขึ้น 4 ลูกบาศก์เซนติเมตรสำหรับถุงเป้ใบที่สาม ซึ่งวัตถุนี้มีมูลค่า 15 บาท ถ้าเลือกวัตถุชิ้นที่สามจะทำให้มี

อุณหภูมิเพิ่มขึ้น 5 องศาเซลเซียสสำหรับถุงใบแรก มีน้ำหนักเพิ่มขึ้น 4 กิโลกรัมสำหรับถุงใบที่สอง มีปริมาตรเพิ่มขึ้น 2 ลูกบาศก์เซนติเมตรสำหรับถุงใบที่สาม ซึ่งวัตถุดิบนี้มีมูลค่า 6 บาท ต้องการเลือกวัตถุให้มีมูลค่ารวมกันแล้วมากที่สุดโดยที่ถุงเป้ทั้ง 3 ถุงต้องไม่ขาด ซึ่งปัญหาถุงเป้แบบหลายมิตินี้สามารถแสดงให้อยู่ในสมการได้ดังนี้

$$\text{Maximize } \sum_{i=1}^n p_i x_i \quad x_i \in \{0,1\} \quad (\text{ข.1})$$

$$\text{Object } \sum_{i=1}^n w_{ij} x_i \leq C_j \quad x_i \in \{0,1\} \quad (\text{ข.2})$$



รูปที่ ข.1 ตัวอย่างปัญหาถุงเป้แบบหลายมิติ

ข.2 การประยุกต์ BPSO กับปัญหาถุงเป้แบบหลายมิติ

เจมส์ เคนเนดี และ รัสเซล อีเบอฮาท [102] ได้นำเสนอขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคแบบไบนารี (A Binary Particle Swarm Optimization หรือ BPSO) เพื่ออนุญาตให้ PSO สามารถนำมาประยุกต์ใช้แก้ไขปัญหาในรูปแบบไบนารี ตั้งแต่นั้นมา BPSO ที่ เจมส์ เคนเนดี และ รัสเซล อีเบอฮาท นำเสนอได้ถูกกำหนดเป็นมาตรฐานของ PSO ที่จะนำมาประยุกต์ใช้แก้ไขปัญหาในรูปแบบไบนารี สำหรับขั้นตอนวิธีการทำงานของ BPSO มีรายละเอียดดังนี้

ข.2.1 กระบวนการกำหนดค่าเริ่มต้น

กระบวนการกำหนดค่าเริ่มต้นให้กับอนุภาคของ BPSO เริ่มต้นจากการแทนรูปแบบคำตอบของปัญหาให้อยู่ในรูปแบบไบนารีและอนุภาคจะอยู่ในรูปแบบไบนารีสตริง (Binary String) ซึ่งจำนวนมิติของอนุภาคจะมีความยาวเท่ากับจำนวนของวัตถุที่เป็นคำตอบของปัญหา เช่น มีวัตถุให้เลือก 10 ชิ้นจำนวนมิติก็จะมี 10 มิติ เป็นต้น สำหรับแต่ละมิติของอนุภาคจะมีค่าเป็นการตัดสินใจ เช่น 0 หรือ 1 ใช้หรือไม่ใช้ เลือกหรือไม่เลือก ถูกหรือผิด เป็นต้น

สำหรับการสร้างประชากรเริ่มต้น เริ่มต้นโดยจะทำการสร้างจำนวนอนุภาคขึ้นมาเท่ากับจำนวนประชากรที่ได้กำหนดเอาไว้ จากนั้นจึงทำการสุ่มกำหนดค่าในแต่ละมิติบนอนุภาคแต่ละตัวด้วยความน่าจะเป็นเริ่มต้นของการเลือก ทำให้มิติบนอนุภาคแต่ละอนุภาคมีค่าเป็น 0 กับ 1 ดังรูป

ข.2 และทำการสุ่มค่าความเร็วของอนุภาคแต่ละตัวโดยจะอยู่ในช่วงระหว่างค่าความเร็วสูงสุดและต่ำสุด สำหรับความเร็วของอนุภาคจะถูกสุ่มให้มีค่าเป็นจำนวนจริงดังรูป ข.2 และความเร็วนี้จะต้องไม่เกินค่าของความเร็วสูงสุดที่กำหนดเอาไว้ หากความเร็วที่สุ่มได้มีค่ามากกว่าค่าความเร็วสูงสุดที่กำหนดเอาไว้ ให้ความเร็วนั้นเปลี่ยนมาใช้ค่าความเร็วสูงสุดที่กำหนดเอาไว้แทน หลังจากนั้นทำการสร้าง PBEST โดยให้เหมือนกับอนุภาคของตนเองดังรูป ข.2 สำหรับ GBEST อนุภาคตัวใดที่มีค่าความเหมาะสมที่สุดก็จะกำหนด GBEST ให้เหมือนกับอนุภาคนั้น

Velocity Dimension	0.8769	0.4352	0.3367	0.5351	0.9367	} 1 Particle
Particle Dimension	1	0	0	1	0	
PBEST Dimension	1	0	0	1	0	
GBEST Dimension	0	1	0	1	0	

รูปที่ ข.2 ตัวอย่างการกำหนดค่าเริ่มต้นกับประชากร

ข.2.2 กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค

กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค (Evaluate) เป็นกระบวนการคำนวณค่าความเหมาะสม ณ ตำแหน่งปัจจุบันของแต่ละอนุภาค โดยค่าความเหมาะสมเหล่านี้จะถูกใช้ในการปรับค่า PBEST และ GBEST ในขั้นตอนต่อไป นอกจากนี้ขั้นตอนนี้ได้ทำการตรวจสอบค่าความเหมาะสมของอนุภาคได้เคลื่อนที่ไปพบคำตอบหรือไม่ ถ้าการค้นหาพบคำตอบแล้วหรือจำนวนรอบการค้นหาคำตอบเท่ากับจำนวนรอบการค้นหาคำตอบสูงสุดที่กำหนดเอาไว้ก็จะจบการค้นหา นอกจากนี้ก็จะดำเนินขั้นตอนต่อไป

ข.2.3 กระบวนการเปลี่ยนแปลง PBEST

กระบวนการเปลี่ยนแปลง PBEST (Update PBEST) เป็นการเปลี่ยนแปลง PBEST โดยใช้ฟังก์ชันคำนวณค่าความเหมาะสม คำนวณหาค่าความเหมาะสมในตำแหน่งที่อนุภาคอยู่ปัจจุบัน หากค่าความเหมาะสมที่คำนวณได้นั้นมีค่าดีกว่า PBEST ก็ให้ PBEST เปลี่ยนมาใช้ค่าใหม่แทน สำหรับ PBEST นี้จะใช้ในการคำนวณความเร็วและทิศทางใหม่ให้กับอนุภาค เพื่อใช้ในการเคลื่อนที่อนุภาคเหล่านั้นไปยังตำแหน่งใหม่

ข.2.4 กระบวนการเปลี่ยนแปลง GBEST

กระบวนการเปลี่ยนแปลง GBEST (Update GBEST) เป็นการเปลี่ยนแปลง GBEST โดยใช้ฟังก์ชันคำนวณค่าความเหมาะสม คำนวณหาค่าความเหมาะสมในตำแหน่งที่อนุภาคอยู่ปัจจุบัน หากค่าความเหมาะสมที่คำนวณได้นั้นมีค่าดีกว่า GBEST ก็ให้ GBEST เปลี่ยนมาใช้ค่าใหม่แทน ซึ่ง GBEST จะทำการเปรียบเทียบกับค่าความเหมาะสมของอนุภาคทุกตัว สำหรับ GBEST นี้จะใช้ในการคำนวณความเร็วและทิศทางใหม่ให้กับอนุภาคทั้งหมด เพื่อใช้เคลื่อนที่อนุภาคเหล่านั้นไปยังตำแหน่งใหม่ สำหรับ GBEST คือคำตอบของการค้นหาด้วย BPSO

ข.2.5 กระบวนการคำนวณความเร็ว

กระบวนการคำนวณความเร็ว เป็นกระบวนการที่ทำการคำนวณความเร็วของอนุภาคแต่ละตัว เพื่อใช้ในการเคลื่อนที่อนุภาคแต่ละตัวในแต่ละรอบ ซึ่งความเร็วนี้จะถูกเปลี่ยนแปลงโดยใช้ข้อมูลของความเร็วเดิมของอนุภาคแต่ละตัว PBEST และ GBEST โดยมีสมการในการคำนวณดังนี้

$$v'_{id} = v_{id} + \eta_1 rand() (P_{id} - X_{id}) + \eta_2 rand() (P_{gd} - X_{id}) \quad (ข.3)$$

ข.2.6 กระบวนการเปลี่ยนตำแหน่ง

สำหรับความเร็วของอนุภาคใน BPSO มีค่าเป็นเซตของจำนวนจริงและตำแหน่งของอนุภาคมีค่าเป็นเซตของบิต ดังนั้นความเร็วของอนุภาคต้องถูกเปลี่ยนเป็นรูปแบบของเซตความน่าจะเป็นโดยใช้ฟังก์ชันซิกมอด (Sigmoid Function) โดยแสดงในสมการ ข.4 สำหรับขั้นตอนการปรับปรุงตำแหน่งของอนุภาคนั้น คือความเร็วของอนุภาคเป็นตัวระบุโอกาสที่ตำแหน่งของอนุภาคหรือบิตจะถูกเลือกหนึ่งหรือศูนย์ โดยขั้นตอนการปรับปรุงตำแหน่งของอนุภาคจะเป็นไปตามสมการ ข.5 โดย $sig(v'_{id})$ คือฟังก์ชันซิกมอดเพื่อใช้สำหรับการแปลงความเร็วของอนุภาคให้อยู่ในรูปความน่าจะเป็น

$$sig(v'_{id}) = \frac{1}{1 + e^{-v'_{id}}} \quad , e \approx 2.7182818 \quad (ข.4)$$

$$X'_{id} = \begin{cases} 1 & \text{if } rand() < sig(v'_{id}) \\ 0 & \text{Otherwise} \end{cases} \quad (ข.5)$$

หลังจากกระบวนการเปลี่ยนตำแหน่ง ก็จะวนกลับไปทำกระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาคในรอบถัดไป สำหรับข้อเสียหลักของ BPSO ถูกกล่าวไว้ในเอกสาร [118] ซึ่งสามารถสรุปได้ว่า ปัญหาแรกของ BPSO คือพฤติกรรมของการปรับปรุงความเร็วของอนุภาคและค่าความเฉลี่ยใน BPSO แตกต่างจาก PSO ปัญหาที่สองของ BPSO คือการปรับปรุงตำแหน่งของ BPSO ใช้เพียงแค่อัตราความเร็วของอนุภาคในรอบปัจจุบันเป็นเกณฑ์ในการปรับปรุงตำแหน่งเท่านั้น แต่ PSO ใช้ความเร็วของอนุภาคในรอบปัจจุบันร่วมกับตำแหน่งของอนุภาคปัจจุบันเพื่อปรับปรุงตำแหน่งของอนุภาค

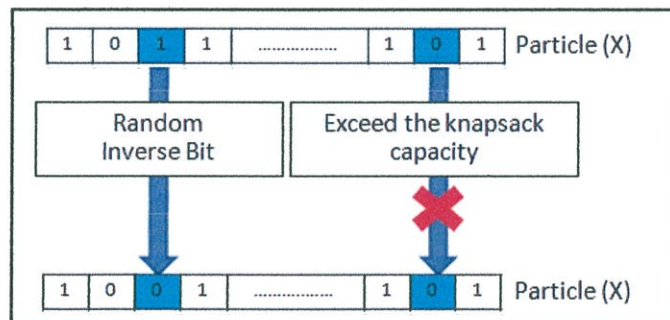
สำหรับการนำ PSO มาประยุกต์ใช้กับปัญหาถุงเป้แบบหลายมิตินั้นจำเป็นต้องเพิ่มเงื่อนไข เนื่องจากปัญหาถุงเป้แบบหลายมิติมีเงื่อนไขว่า สิ่งของที่เลือกเมื่อใส่ลงไปถุงเป้แล้ว ถุงเป้ต้องสามารถรับน้ำหนักสิ่งของเหล่านั้นได้ทั้งหมด โดยที่วัตถุหรือสิ่งของที่ถูกเลือกจะถูกแทนในรูปแบบของบิตหรือบิต ถ้าวัตถุถูกเลือกจะมีค่าบิตเป็น 1 ในทางกลับกันถ้าวัตถุไม่ถูกเลือกจะมีค่าบิตเป็น 0 สำหรับ PSO จะถูกแทนที่ด้วย BPSO และถ้าการเปลี่ยนบิตจาก 0 เป็น 1 ทุกครั้งจะถูกตรวจสอบความจุของถุงเป้ทั้งหมดว่าสามารถรองรับ น้ำหนักได้หรือไม่ ถ้าสามารถรองรับ น้ำหนักได้จึงจะเปลี่ยนค่าจาก 0 เป็น 1 ถ้าไม่สามารถรองรับ น้ำหนักได้ก็จะคงค่าบิตเดิมเอาไว้ ซึ่งการตรวจสอบความจุของถุงเป้ทั้งหมดเป็นการรับประกันว่าการค้นหาคำตอบจะไม่เกินขอบเขตของการค้นหาคำตอบหรือปริภูมิของคำตอบ

ข.3 การประยุกต์ NCM และรีโพซิชั่นกับ BPSO กับปัญหาถุงเป้แบบหลายมิติ

การประยุกต์การกลายพันธุ์ประเภท NCM คือการสุ่มเปลี่ยนค่าตำแหน่งของอนุภาคใดๆ ภายในกลุ่มของประชากร โดยเลือกเปลี่ยนค่าในบางมิติของอนุภาค ในจำนวนเพียงเล็กน้อย โดยใช้พื้นฐานการทำงานเหมือนกับหัวข้อ 3.1 แต่ต้องมีการปรับวิธีการทำงานเนื่องจากปัญหาในรูปแบบไบนารีซึ่งเป็นปัญหาที่ไม่มีทิศทางและขนาด โดยในวิทยานิพนธ์ฉบับนี้เลือกใช้การกลายพันธุ์แบบกลับบิต (Inverse Bit Mutation) โดยวิธีการทำงานจะทำการสุ่มเปลี่ยนค่าบิตในอนุภาคแต่ละอนุภาค โดยทำการเปลี่ยนค่าจาก 0 เป็น 1 และจาก 1 เป็น 0 ด้วยความน่าจะเป็นในการกลายพันธุ์ซึ่งกำหนดเอาไว้ก่อนหน้า ดังแสดงตัวอย่างในรูป ข.3

สำหรับวิธีรีโพซิชั่นได้เลือกใช้การกลายพันธุ์แบบกลับบิตเช่นเดียวกับการประยุกต์การกลายพันธุ์ประเภท NCM แต่เนื่องจากการปรับปรุงตำแหน่งของ BPSO นั้นขึ้นอยู่กับความเร็วของอนุภาค ดังนั้นการรีเซ็ตตำแหน่งของอนุภาคจึงจำเป็นต้องรีเซ็ตความเร็วของอนุภาคด้วยเช่นกันไม่เช่นนั้นอนุภาคก็จะเปลี่ยนตัวเองกลับมาเป็นตำแหน่งเดิมก่อนรีเซ็ต เนื่องจากได้รับอิทธิพลจากความเร็วของอนุภาคทำให้การรีเซ็ตจึงไม่เกิดประโยชน์

สำหรับการนำการประยุกต์การกลายพันธุ์ประเภท NCM และรีโพซิชั่นไปประยุกต์ใช้กับปัญหาถุงเป้แบบหลายมิติ การที่จะกลับบิตจาก 0 เป็น 1 ทุกครั้งจะถูกตรวจสอบความจุของถุงเป้ทั้งหมดว่าสามารถรองรับ น้ำหนักได้หรือไม่ ถ้าหากถุงเป้ไม่สามารถรองรับ น้ำหนักของวัตถุทั้งหมดได้ก็จะคงค่าบิตเดิมเอาไว้



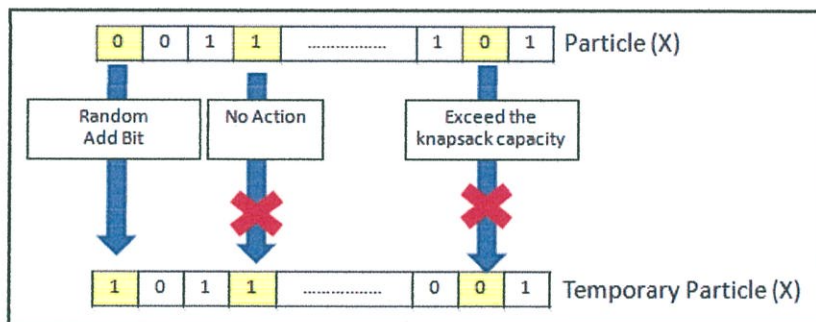
รูปที่ ข.3 ตัวอย่างการทำงานของกระบวนการกลายพันธุ์ประเภท NCM

ข.4 การประยุกต์ CM กับ BPSO กับปัญหาถุงเป้แบบหลายมิติ

การประยุกต์การกลายพันธุ์ประเภท CM คือการสุ่มเปลี่ยนค่าตำแหน่งของอนุภาคใดๆ ภายในกลุ่มของประชากร โดยเลือกเปลี่ยนค่าในบางมิติของอนุภาคในจำนวนเพียงเล็กน้อย โดยใช้พื้นฐานการทำงานเหมือนกับหัวข้อ 3.1 แต่ต้องมีการปรับวิธีการทำงานเนื่องจากปัญหาในรูปแบบไบนารีซึ่งเป็นปัญหาที่ไม่มีทิศทางและขนาด โดยในวิทยานิพนธ์ฉบับนี้เลือกใช้การกลายพันธุ์แบบเพิ่มบิต (Add Bit Mutation) โดยวิธีการทำงานจะทำการสุ่มเปลี่ยนค่าบิตในอนุภาคแต่ละอนุภาค โดยทำการเปลี่ยนค่าจาก 0 เป็น 1 เท่านั้น ด้วยความน่าจะเป็นในการกลายพันธุ์ซึ่งกำหนดเอาไว้ก่อนหน้า ถ้าหากพบตำแหน่งที่บิตมีค่าเป็น 1 ก็จะคงค่าบิตเดิมเอาไว้ดังแสดงตัวอย่างในรูป ข.4

เนื่องจากโดยปกติถ้าหากเลือกใช้การกลายพันธุ์แบบกลับบิตกับปัญหาถุงเป้แบบหลายมิติ โดยส่วนมากแล้วคำตอบที่ได้จากการกลายพันธุ์มักจะแย่กว่าตำแหน่งที่ถูกดำเนินการกลายพันธุ์ ซึ่ง CM จะไม่นำผลจากการกลายพันธุ์มาใช้ ดังนั้นการดำเนินการกลายพันธุ์จึงสูญเปล่า เพื่อให้การกลายพันธุ์เป็นส่วนที่ช่วยปรับปรุงคำตอบให้ BPSO จึงได้ออกแบบให้ดำเนินการกลายพันธุ์โดยสุ่มเปลี่ยนบิตจาก 0 เป็น 1 เท่านั้น ถ้าหากสามารถเปลี่ยนได้ผลลัพธ์ที่ได้นั้นก็ยังสามารถนำใช้แทนที่ตำแหน่งที่ถูกดำเนินการกลายพันธุ์ได้ การกลายพันธุ์จึงมีประโยชน์และช่วยปรับปรุงคำตอบได้

อย่างไรก็ตามการประยุกต์การกลายพันธุ์ CM กับปัญหาถุงเป้แบบหลายมิติ การที่จะกลับบิตจาก 0 เป็น 1 ทุกครั้งจะถูกตรวจสอบความจุของถุงเป้ทั้งหมดว่าสามารถรองรับ น้ำหนักได้หรือไม่ ถ้าหากถุงเป้ไม่สามารถรองรับ น้ำหนักของวัตถุทั้งหมดได้ ก็จะคงค่าบิตเดิมเอาไว้เช่นเดียวกับ NCM



รูปที่ ข.4 ตัวอย่างการทำงานของกระบวนการกลายพันธุ์ประเภท CM

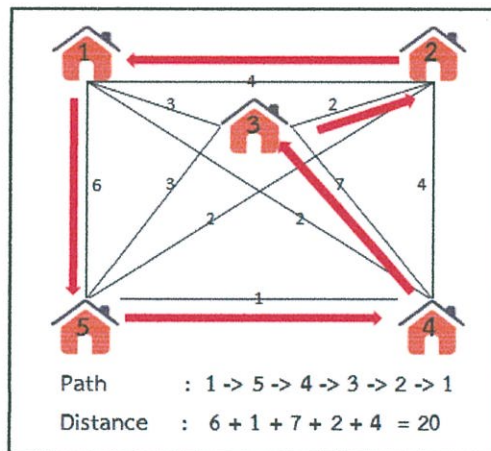
ภาคผนวก ค

การประยุกต์ PSO กับปัญหาการเดินทางของพนักงานขาย

ปัญหาการเดินทางของพนักงานขาย คือปัญหาการหาจุดที่ดีที่สุดที่มีจุดประสงค์ต้องการหาเส้นทางที่สั้นที่สุดที่ทำให้พนักงานขายสามารถเดินทางไปพบลูกค้า (Nodes หรือ โหนด) ได้ครบทุกคน โดยพนักงานขายจะเดินทางไปพบลูกค้าเพียงคนละหนึ่งครั้งเท่านั้น และเมื่อเดินทางพบลูกค้าครบทุกคนพนักงานขายจะต้องเดินทางกลับมายังจุดเริ่มต้น

ค.1 รูปแบบคำตอบของปัญหาการเดินทางของพนักงานขาย

ปัญหาการเดินทางของพนักงานขายเป็นปัญหาในรูปแบบการสลับลำดับกันในพีชคณิต ซึ่งคำตอบมีลักษณะเป็นลำดับหมายเลขของโหนดที่เรียกต่อกัน โดยลำดับหมายเลขของโหนดจะไม่มีการซ้ำกัน เพราะพนักงานขายจะเดินทางไปพบลูกค้าเพียงคนละหนึ่งครั้งเท่านั้น และลำดับหมายเลขของโหนดคือเส้นทางการเดินทางของพนักงานขาย ตัวอย่างเช่นลำดับหมายเลขของโหนด คือ 1, 5, 4, 3, 2 หมายความว่า การเดินทางของพนักงานขายต้องเดินทางเริ่มต้นจากโหนดที่ 1 ไปโหนดที่ 5 ไปโหนดที่ 4 ไปโหนดที่ 3 ไปโหนดที่ 2 และกลับมาโหนดที่ 1 ดังรูป ค.1



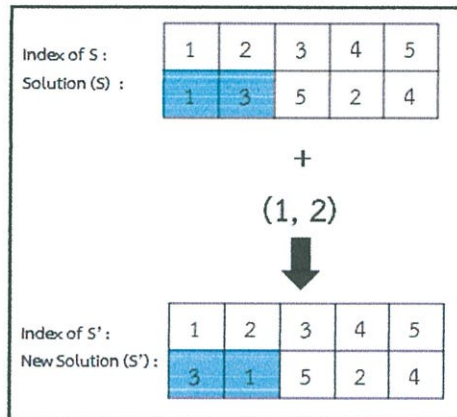
รูปที่ ค.1 ตัวอย่างรูปแบบคำตอบของ TSP

ค.2 ขั้นตอนการสลับที่คำตอบของปัญหาการเดินทางของพนักงานขาย

คำตอบของปัญหาการเดินทางของพนักงานขาย คือการเรียงลำดับจำนวน n โหนด โดยใช้สัญลักษณ์เป็น $S = (N_1, N_2, N_3, \dots, N_n)$ โดยที่ N คือ โหนด และ S คือ คำตอบของปัญหาการเดินทางของพนักงานขาย

สำหรับการสลับลำดับโหนดใช้สัญลักษณ์เป็น $SO(i_1, i_2)$ หรือคู่อันดับการสลับที่ หมายความว่า ให้สลับโหนดที่อยู่ในดัชนี i_1 กับโหนดที่อยู่ในดัชนี i_2 เมื่อนำมาเขียนอยู่ในรูปสมการคือ $S' = S + SO(1, 2)$ จากสมการนี้มีเครื่องหมาย "+" ซึ่งหมายความว่า ให้ดำเนินการสลับที่โหนดที่อยู่ใน S โดยสลับที่โหนดที่อยู่ในดัชนี 2 กับโหนดที่อยู่ในดัชนี 1 แล้วทำให้เกิดคำตอบใหม่ขึ้น (S') ตัวอย่างเช่นกำหนดให้ $S = (1,$

3, 5, 2, 4) และเมื่อดำเนินการสลับที่โหนด $S' = S + SO(1, 2) = (1, 3, 5, 2, 4) + (1, 2)$ ผลลัพธ์จากการดำเนินการสลับที่โหนด คือ $S' = (3, 1, 5, 2, 4)$ ดังรูป ค.2



รูปที่ ค.2 ตัวอย่างการดำเนินการสลับที่คำตอบของ TSP

สำหรับการสลับที่โหนดนั้นสามารถดำเนินการสลับที่ได้หลายครั้ง ดังนั้นคู่อันดับการสลับที่สามารถเขียนในรูปแบบเซตคือ $SS = (SO_1, SO_2, SO_3, \dots, SO_n)$ โดย SO แต่ละตัวคือคู่อันดับการสลับที่ ดังนั้นเมื่อคำตอบถูกดำเนินการสลับที่หลายครั้งสามารถเขียนเป็นสมการได้ดังนี้ $S' = S + SS = S + (SO_1, SO_2, SO_3, \dots, SO_n)$

สำหรับกรณีการนำคู่อันดับการสลับที่มารวมกันนั้น จะมีสัญลักษณ์เป็น \oplus ซึ่งหมายความว่าให้นำคู่อันดับการสลับที่สองชุดมารวมกัน ตัวอย่างเช่น $SS_1 = [(1, 2), (3, 4)]$, $SS_2 = [(2, 5), (4, 1)]$ เมื่อนำคู่อันดับการสลับที่มารวมกันจะเขียนสมการได้ดังนี้ $SS' = SS_1 \oplus SS_2$ และผลลัพธ์จากการดำเนินการคือ $SS' = [(1, 2), (3, 4), (2, 5), (4, 1)]$

สำหรับกรณีที่นำคำตอบจากปัญหาการเดินทางของพนักงานขายชุดเดียวกัน (จำนวนโหนดเท่ากันและมีตัวเลขของโหนดชุดเดียวกัน) มาลบกันจะทำให้เกิดคู่อันดับการสลับที่ โดยสัญลักษณ์การลบกันระหว่างคำตอบคือ "-" โดยสามารถเขียนสมการได้ดังนี้ $SS = S_1 - S_2$ ถ้าหากย้ายข้างสมการจะได้ผลลัพธ์คือ $S_1 = S_2 + SS$ ตัวอย่างเช่น กำหนดให้ $S_1 = (1, 2, 3, 4, 5)$ และ $S_2 = (2, 3, 1, 5, 4)$ ถ้านำ $SS = S_1 - S_2$ หรือเท่ากับ $S_1 = S_2 + SS$ หมายความว่าดำเนินการสลับลำดับโหนดให้ S_2 กลายเป็น S_1 โดยใช้คู่อันดับการสลับที่ SS ซึ่งนั่นคือผลลัพธ์การลบกันของ S_1 กับ S_2 สำหรับ SS ได้ถูกสร้างขึ้นมาโดยพิจารณาจากหลักการดังนี้

- โหนดที่อยู่ในดัชนีของ $S_1(1)$ เท่ากับโหนดที่อยู่ในดัชนีของ $S_2(3)$ ดังนั้นการสลับที่ชุดแรกคือ 1 กับ 3 หรือ $SO(1,3)$ สามารถเขียนออกเป็นสมการคือ $S'_2 = S_2 + SO(1,3)$ ซึ่งจะได้ผลลัพธ์คือ 1, 3, 2, 5, 4 ดังรูป ค.3
- หลังจากสลับที่ครั้งแรกแล้วพบว่า ผลลัพธ์ที่ได้เมื่อเปรียบเทียบกับ S_1 พบว่าโหนดที่อยู่ในดัชนีของ $S_1(2)$ เท่ากับโหนดที่อยู่ในดัชนีของ $S'_2(3)$ ดังนั้นการสลับที่ชุดสองคือ 2 กับ 3 หรือ $SO(2,3)$ สามารถเขียนออกเป็นสมการคือ $S''_2 = S'_2 + SO(2,3)$ ซึ่งจะได้ผลลัพธ์คือ 1, 2, 3, 5, 4 ดังรูป ค.3

- หลังจากสลับที่ครั้งสองแล้วพบว่า ผลลัพธ์ที่ได้เมื่อเปรียบเทียบกับ S_1 พบว่าโนดที่อยู่ในดัชนีของ $S_1(4)$ เท่ากับโนดที่อยู่ในดัชนีของ $S''_2(5)$ ดังนั้นการสลับที่ชุดสองคือ 4 กับ 5 หรือ $SO(4, 5)$ สามารถเขียนออกเป็นสมการคือ $S'''_2 = S''_2 + SO(4,5)$ ซึ่งจะได้ผลลัพธ์คือ 1, 2, 3, 4, 5 ซึ่งผลลัพธ์ที่ได้นั้นคือเซตลำดับของ S_1 ดังรูป ค.3 เมื่อเขียนสรุปผลการดำเนินงานทั้งหมดคือ $SS = S_1 - S_2 = [SO(1,3), SO(2,3), SO(4,5)]$ ซึ่ง SS ผลลัพธ์จากการลบกันของ S_1 และ S_2

Index of S_2 :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5	+ $SO(1,3) = S'_2$
1	2	3	4	5			
Solution (S_2) :	<table border="1"><tr><td>2</td><td>3</td><td>1</td><td>5</td><td>4</td></tr></table>	2	3	1	5	4	
2	3	1	5	4			
Solution (S_1) :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5	
1	2	3	4	5			
Index of S'_2 :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5	+ $SO(2,3) = S''_2$
1	2	3	4	5			
Solution (S'_2) :	<table border="1"><tr><td>1</td><td>3</td><td>2</td><td>5</td><td>4</td></tr></table>	1	3	2	5	4	
1	3	2	5	4			
Solution (S_1) :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5	
1	2	3	4	5			
Index of S''_2 :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5	+ $SO(4,5) = S_1$
1	2	3	4	5			
Solution (S''_2) :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>5</td><td>4</td></tr></table>	1	2	3	5	4	
1	2	3	5	4			
Solution (S_1) :	<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr></table>	1	2	3	4	5	
1	2	3	4	5			

รูปที่ ค.3 ตัวอย่างการดำเนินการลบกันของคำตอบของ TSP

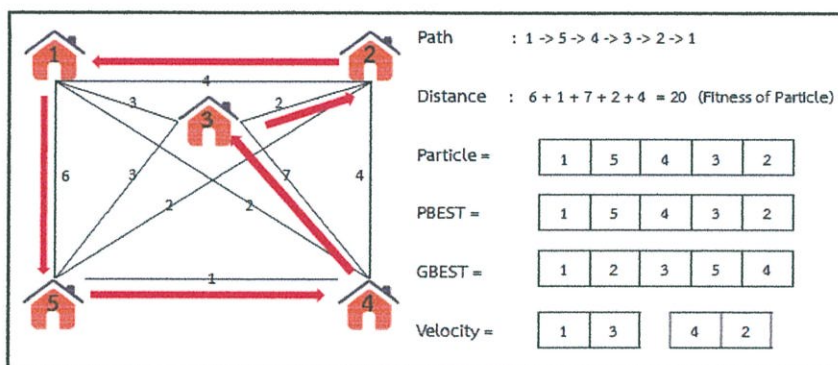
ค.3 การประยุกต์ PPSO กับปัญหาการเดินทางของพนักงานขาย

เนื่องจาก PSO ถูกออกแบบมาสำหรับการแก้ปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่อง จึงไม่สามารถนำมาประยุกต์ใช้แก้ปัญหาการเดินทางของพนักงานขายได้โดยตรง เพื่อให้ PSO สามารถนำมาใช้แก้ปัญหาการเดินทางของพนักงานขายได้ จึงต้องดำเนินการปรับปรุงขั้นตอนการทำงานของ PSO [105, 106] และเรียก PSO ที่ถูกปรับปรุงขั้นตอนการทำงานนี้ว่า วิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาคในรูปแบบการสลับลำดับ (Permutation Particle Swarm Optimization หรือ PPSO) สำหรับขั้นตอนวิธีการทำงานของ PPSO มีรายละเอียดดังนี้

ค.3.1 กระบวนการกำหนดค่าเริ่มต้น

กระบวนการกำหนดค่าเริ่มต้นให้กับอนุภาคของ PPSO นั้นเริ่มต้นโดยการแทนรูปแบบคำตอบของปัญหาให้อยู่ในรูปแบบการสลับลำดับกันในพีชคณิต ดังนั้นอนุภาคของ PPSO แต่ละอนุภาคจะมีมิติที่เป็นลำดับของโนด เช่น 1, 5, 4, 3, 2 หมายความว่า การเดินทางของพนักงานขายต้องเดินทางเริ่มต้นจากโนดที่ 1 ไปโนดที่ 5 ไปโนดที่ 4 ไปโนดที่ 3 ไปโนดที่ 2 และกลับมาโนดที่ 1 ดังรูป ค.4 สำหรับ PBEST ของแต่ละอนุภาคก็จะมีมิติที่เก็บลำดับของโนดเช่นเดียวกับอนุภาค ดังรูป ค.4 สำหรับ GBEST จะมีมิติที่เก็บลำดับของโนดเช่นเดียวกับอนุภาค ดังรูป ค.4 สำหรับความเร็วของแต่ละอนุภาคก็จะมีมิติที่เก็บเซตคู่อันดับการสลับที่ ดังรูป ค.4

สำหรับการสร้างประชากรเริ่มต้นนั้นจะทำการสร้างจำนวนอนุภาคขึ้นมาเท่ากับจำนวนประชากรที่ได้กำหนดเอาไว้ จากนั้นจะดำเนินการสุ่มหมายเลขของโนดซึ่งเป็นค่าในแต่ละมิติบนอนุภาคแต่ละตัว โดยภายในอนุภาคเดียวกันต้องไม่มีหมายเลขของโนดซ้ำกัน ทำให้มิติบนอนุภาคแต่ละอนุภาคมีค่าเป็นหมายเลขของโนด ดังรูป ค.4 และทำการสุ่มค่าความเร็วของแต่ละอนุภาคโดยสุ่มหมายเลขเป็นคู่อันดับการสลับที่ ดังรูป ค.4 โดยจำนวนคู่อันดับสลับที่นี้ต้องมีจำนวนคู่อันดับไม่เกินจำนวนคู่อันดับการสลับที่ของความเร็วสูงสุดที่กำหนดเอาไว้ ถ้าหากจำนวนคู่อันดับสลับที่เกินกว่าจำนวนคู่อันดับการสลับที่ของความเร็วสูงสุดที่ได้กำหนดเอาไว้ จำนวนคู่อันดับสลับที่เกินจะไม่นำมาใช้ หลังจากนั้นทำการสร้าง PBEST โดยให้เหมือนกับอนุภาคของตนเองดังรูป ค.4 สำหรับ GBEST อนุภาคตัวใดที่มีค่าความเหมาะสมที่สุดก็จะกำหนด GBEST ให้เหมือนกับอนุภาคตัวนั้น ดังรูป ค.4



รูปที่ ค.4 ตัวอย่างอนุภาคของ PPSO ที่นำมาประยุกต์ใช้กับ TSP

ค.3.2 กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค

กระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาค (Evaluate) เป็นกระบวนการคำนวณค่าความเหมาะสม ณ ตำแหน่งปัจจุบันของแต่ละอนุภาค สำหรับการคำนวณค่าความเหมาะสมของแต่ละอนุภาคจะคำนวณจากระยะการเดินทางตามลำดับของโนดที่ได้จัดเรียงไว้ในอนุภาคแต่ละตัว ซึ่งจุดประสงค์ของ TSP คือต้องการจัดเรียงให้ค่าความเหมาะสมมีค่าน้อยที่สุด ตัวอย่างเช่นรูป ค.4 จะพบความระยะทางโดยรวมของอนุภาคนี้คือ 20 กิโลเมตร ซึ่งค่านี้เป็นค่าความเหมาะสมของอนุภาคตัวนี้ นอกจากนี้ขั้นตอนนี้ได้ทำการตรวจสอบค่าความเหมาะสมของอนุภาคได้เคลื่อนที่ไปพบคำตอบหรือไม่ ถ้าการค้นหาคำตอบแล้วหรือจำนวนรอบการค้นหาคำตอบเท่ากับจำนวนรอบการค้นหาคำตอบสูงสุดที่กำหนดเอาไว้ก็จะจบการค้นหา นอกจากนั้นก็ดำเนินขั้นตอนต่อไป

ค.3.3 กระบวนการเปลี่ยนแปลง PBEST

กระบวนการเปลี่ยนแปลง PBEST (Update PBEST) เป็นการเปลี่ยนแปลง PBEST โดยใช้ฟังก์ชันคำนวณค่าความเหมาะสมคำนวณหาค่าความเหมาะสมในตำแหน่งที่อนุภาคอยู่ปัจจุบัน หากค่าความเหมาะสมที่คำนวณได้นั้นมีค่าดีกว่า PBEST ก็ให้ PBEST เปลี่ยนมาใช้ค่าใหม่แทน

สำหรับ PBEST นี้จะใช้ในการคำนวณความเร็วและทิศทางใหม่ให้กับอนุภาค เพื่อใช้เคลื่อนที่อนุภาคเหล่านั้นไปยังตำแหน่งใหม่

ค.3.4 กระบวนการเปลี่ยนแปลง GBEST

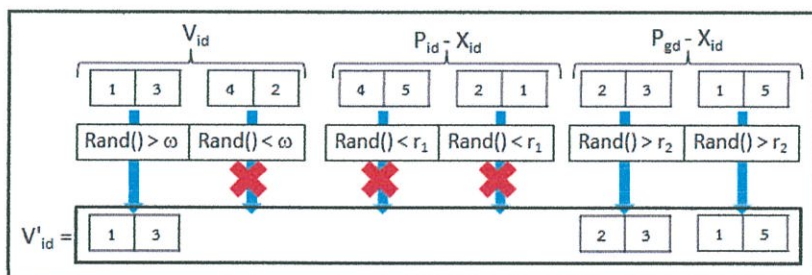
กระบวนการเปลี่ยนแปลง GBEST (Update GBEST) เป็นการเปลี่ยนแปลง GBEST โดยใช้ฟังก์ชันคำนวณค่าความเหมาะสม คำนวณหาค่าความเหมาะสมในตำแหน่งที่อนุภาคอยู่ปัจจุบัน หากค่าความเหมาะสมที่คำนวณได้นั้นมีค่าดีกว่า GBEST ก็ให้ GBEST เปลี่ยนมาใช้ค่าใหม่แทน ซึ่ง GBEST จะทำการเปรียบเทียบกับค่าความเหมาะสมของอนุภาคทุกตัว สำหรับ GBEST นี้จะใช้ในการคำนวณความเร็วและทิศทางใหม่ให้กับอนุภาคทั้งหมด เพื่อใช้เคลื่อนที่อนุภาคเหล่านั้นไปยังตำแหน่งใหม่ สำหรับ GBEST คือคำตอบของการค้นหาด้วย PPSO

ค.3.5 กระบวนการคำนวณความเร็ว

กระบวนการคำนวณความเร็ว เป็นกระบวนการที่ทำการคำนวณความเร็วของอนุภาคแต่ละตัว เพื่อให้อนุภาคแต่ละตัวเกิดการเคลื่อนที่ในแต่ละรอบ ซึ่งความเร็วนี้จะถูกเปลี่ยนแปลงโดยใช้ข้อมูลของความเร็วเดิมของอนุภาคแต่ละตัว PBEST และ GBEST โดยมีสมการในการคำนวณดังนี้

$$V'_{id} = \omega \cdot V_{id} \oplus r_1 \cdot (P_{id} - X_{id}) \oplus r_2 \cdot (P_{gd} - X_{id}) \quad r_1, r_2, \omega \in [0,1] \quad (ค.1)$$

โดย $P_{id} - X_{id}$ และ $P_{gd} - X_{id}$ คือขั้นตอนการลบกันระหว่างคำตอบ ซึ่งการดำเนินการนั้นได้ถูกกล่าวไว้ในหัวข้อ ค.2 สำหรับเครื่องหมาย "⊕" คือการรวมกันของคู่อันดับการสลับที่ซึ่งการดำเนินการนั้นได้ถูกกล่าวไว้ในหัวข้อ ค.2 สำหรับ $\omega \cdot V_{id}$ และ $r_1 \cdot (P_{id} - X_{id})$ และ $r_2 \cdot (P_{gd} - X_{id})$ หมายความว่าถ้าหากค่าสุ่มในช่วง 0 ถึง 1 มีค่ามากกว่า ω และ r_1 และ r_2 ตามลำดับ คู่อันดับการสลับที่ที่อยู่ในเซตของ V_{id} และ $(P_{id} - X_{id})$ และ $(P_{gd} - X_{id})$ ตามลำดับ จะถูกดำเนินการ ⊕ ซึ่งจะทำให้คู่อันดับการสลับที่นั้นมาปรากฏอยู่ในเซตของความเร็วปัจจุบัน (V'_{id}) ดังรูป ค.5



รูปที่ ค.5 ตัวอย่างกระบวนการคำนวณความเร็วของ PPSO

ค.3.6 กระบวนการเปลี่ยนตำแหน่ง

กระบวนการเปลี่ยนตำแหน่ง (Update Position) คือ การปรับปรุงตำแหน่งของอนุภาคใดๆ โดยอนุภาคนั้นๆ จะเคลื่อนที่จากตำแหน่งเดิมไปยังตำแหน่งใหม่ ด้วยระยะที่คำนวณได้จากค่าความเร็วของอนุภาคนั้นๆ โดยมีสมการดังนี้

$$X_{id}' = X_{id} + V_{id}' \quad (\text{ค.2})$$

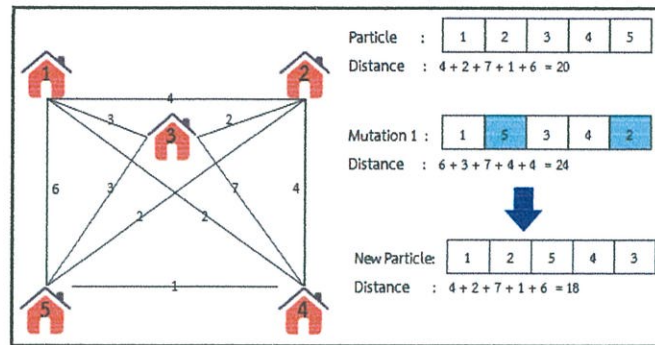
จากสมการ ค.2 หมายความว่าให้ดำเนินการสลับลำดับโนดของ X_{id} ตามลำดับที่อยู่ ใน V_{id}' ซึ่งทำให้อนุภาคเกิดการเคลื่อนที่ไปยังตำแหน่งใหม่ ซึ่งการดำเนินการสลับลำดับโนดนั้นได้ถูกกล่าวไว้ในหัวข้อ ค.2 หลังจากกระบวนการเปลี่ยนตำแหน่ง ก็จะวนกลับไปทำกระบวนการประเมินค่าตำแหน่งของแต่ละอนุภาคในรอบถัดไป

ค.4 การประยุกต์ NCM และรีโพอิชชันกับ PPSO กับปัญหาการเดินทางของพนักงานขาย

การประยุกต์การกลายพันธุ์ประเภท NCM คือการการสุ่มเปลี่ยนค่าตำแหน่งของอนุภาคใดๆ ภายในกลุ่มของประชากร โดยเลือกเปลี่ยนค่าในบางมิติของอนุภาคในจำนวนเพียงเล็กน้อย โดยใช้พื้นฐานการทำงานเหมือนกับหัวข้อ 3.1 แต่ต้องมีการปรับวิธีการทำงานเนื่องจากปัญหาในรูปแบบการสลับลำดับกันในพื้นที่คิดเป็นปัญหาที่ไม่มีทิศทางและขนาด โดยวิธานิพนธ์ฉบับนี้ได้เลือกใช้การกลายพันธุ์แบบสลับโนด (Swap Node Mutation) โดยรหัสจำลองของการกลายพันธุ์แบบสลับโนดแสดงอยู่ในรูป ค.6 และรูปตัวอย่างแสดงในรูป ค.7 โดย $\text{rand}()$ คือค่าสุ่มในช่วง 0 ถึง 1 PM คือความน่าจะเป็นของการกลายพันธุ์ N คือจำนวนประชากร D คือจำนวนมิติหรือโนดทั้งหมดภายในอนุภาค

For i to N	1
For j to D	2
If $\text{rand}() < \text{PM}$	3
k = ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq j$	4
สลับที่โนด x_{ij} กับ x_{ik}	5
End if	6
End j	7
End i	8

รูปที่ ค.6 รหัสจำลองของการกลายพันธุ์แบบสลับโนด



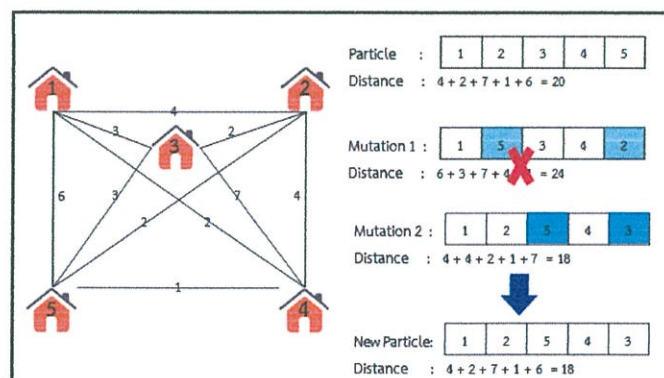
รูปที่ ค.7 ตัวอย่างการทำงานกลายพันธุ์ประเภท NCM กับ TSP

จากรูป ค.7 อนุภาคมีลำดับของโนดเป็น 1, 2, 3, 4, 5 ซึ่งระยะทางรวมที่ได้คือ 20 กิโลเมตร ในการกลายพันธุ์ได้มีการสลับที่โนดในดัชนีที่ 2 กับ 5 ผลลัพธ์จากการกลายพันธุ์ คือ 1, 5, 3, 4, 2 ซึ่งระยะทางรวมที่ได้คือ 24 กิโลเมตร ผลจากการกลายพันธุ์จะนำมาใช้โดยแทนที่อนุภาค

สำหรับวิธีวิเศษชันได้เลือกใช้การกลายพันธุ์แบบสลับโนดเช่นเดียวกับการประยุกต์การกลายพันธุ์ประเภท NCM แต่เนื่องจากการปรับปรุงตำแหน่งของ PPSO นั้นขึ้นอยู่กับความเร็วของอนุภาค เพราะความเร็วของอนุภาคจะเป็นตัวสลับที่ให้อนุภาคนั้นเปลี่ยนกลับเป็นอนุภาคเดิมก่อนรีเซ็ต ทำให้การรีเซ็ตไม่เกิดประโยชน์ ดังนั้นการรีเซ็ตนอกจากรีเซ็ตตำแหน่งของอนุภาคแล้วต้องรีเซ็ตความเร็วของอนุภาคด้วย

ค.5 การประยุกต์ CM กับ PPSO กับปัญหาการเดินทางของพนักงานขาย

การประยุกต์การกลายพันธุ์ประเภท CM คือการการสุ่มเปลี่ยนค่าตำแหน่งของอนุภาคใดๆ ภายในกลุ่มของประชากร โดยเลือกเปลี่ยนค่าในบางมิติของอนุภาคในจำนวนเพียงเล็กน้อย โดยใช้พื้นฐานการทำงานเหมือนกับหัวข้อ 3.1 แต่ต้องมีการปรับวิธีการทำงานเนื่องจากปัญหาในรูปแบบการสลับลำดับกันในพื้นที่ชนิดเป็นปัญหาที่ไม่มีทิศทางและขนาด โดยในวิทยานิพนธ์ฉบับนี้เลือกใช้การกลายพันธุ์แบบสลับโนดเช่นเดียวกับการประยุกต์การกลายพันธุ์ประเภท NCM แต่สำหรับ CM ผลจากการกลายพันธุ์ถูกนำมาใช้ก็ต้องเมื่อผลจากการกลายพันธุ์นั้นดีกว่าตำแหน่งที่ถูกดำเนินการกลายพันธุ์เท่านั้นดังแสดงตัวอย่างในรูป ค.8



รูปที่ ค.8 ตัวอย่างการทำงานกลายพันธุ์ประเภท CM กับ TSP

จากรูป ค.8 อนุภาคมีลำดับของโนตเป็น 1, 2, 3, 4, 5 ซึ่งระยะทางรวมที่ได้คือ 20 กิโลเมตร ในการกลายพันธุ์ครั้งแรกได้มีการสลับที่โนตในดัชนีที่ 2 กับ 5 ผลลัพธ์จากการกลายพันธุ์ คือ 1, 5, 3, 4, 2 ซึ่งระยะทางรวมที่ได้คือ 24 กิโลเมตร เมื่อเปรียบเทียบระยะทางของอนุภาคที่ถูกดำเนินการกลายพันธุ์พบว่า ระยะจากการกลายพันธุ์มีระยะทางที่มากกว่าระยะทางของอนุภาค ดังนั้นผลจากการกลายพันธุ์จึงแยกว่าอนุภาคจึงไม่นำผลนั้นมาใช้ สำหรับการกลายพันธุ์ในรอบที่สอง ได้มีการสลับที่โนตในดัชนีที่ 3 กับ 5 ผลลัพธ์จากการกลายพันธุ์ คือ 1, 2, 5, 4, 3 ซึ่งระยะทางรวมที่ได้คือ 18 กิโลเมตร เมื่อเปรียบเทียบระยะทางของอนุภาคที่ถูกดำเนินการกลายพันธุ์พบว่า ระยะจากการกลายพันธุ์มีระยะทางที่น้อยกว่าระยะทางของอนุภาค ดังนั้นผลจากการกลายพันธุ์จึงดีกว่าอนุภาคจึงนำผลจากการกลายพันธุ์นั้นมาใช้โดยแทนที่อนุภาค

ภาคผนวก ง

การทดลองการประยุกต์การกลายพันธุ์และรีโพอิชชันกับ MKP

ในภาคผนวกนี้จะดำเนินการทดลองเพื่ออธิบายแนวคิดที่มาของขั้นตอนวิธีที่นำเสนอ (MRPSO) เมื่อนำไปประยุกต์ใช้กับปัญหาถุงเป้หลายมิติ ซึ่งเป็นปัญหาในรูปแบบโบนารีจึงมีคุณสมบัติแตกต่างจากปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่องที่ได้ดำเนินการทดลองในบทที่ 5

โดยการทดลองในภาคผนวกนี้จะพิสูจน์ว่าผลลัพธ์จากการค้นหาคำตอบด้วยปัญหาถุงเป้หลายมิติจะให้ผลลัพธ์เหมือนกับฟังก์ชันมาตรฐานในบทที่ 5 สำหรับรายละเอียดขั้นตอนวิธีที่นำมาทดลองถูกกล่าวไว้ในบทที่ 5 และรายละเอียดชุดการทดลองถูกกล่าวไว้ในบทที่ 5 ส่วนการทดลองสามารถแบ่งออกเป็น 6 ส่วน โดยพิจารณาจากจุดประสงค์การทดลองมีรายละเอียดดังนี้

- การทดลองปัญหาการติดในจุดสุดสัมพัทธ์ของ BPSO โดยทดลองกับปัญหาถุงเป้แบบหลายมิติ
- การทดลองการประยุกต์ใช้การกลายพันธุ์กับ BPSO โดยทดลองกับปัญหาถุงเป้แบบหลายมิติ
- การทดลองปัญหาจากการกลายพันธุ์แบบ NCM โดยทดลองกับปัญหาถุงเป้แบบหลายมิติ
- การทดลองปัญหาจากการกลายพันธุ์และแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM โดยทดลองกับปัญหาถุงเป้แบบหลายมิติ
- การทดลองการประยุกต์ใช้รีรันและรีโพอิชชันโดยทดลองกับปัญหาถุงเป้แบบหลายมิติ
- การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพอิชชันโดยทดลองกับปัญหาถุงเป้แบบหลายมิติ

ตารางที่ ง.1 สรุปสมบัติของปัญหาถูกแบ่งแบบหลายมิติที่ใช้ในการทดลอง

ชื่อปัญหา	ค่าที่ดีที่สุด	จำนวนเงื่อนไข	จำนวนวัตถุ
SETO1	7772	30	60
SETO2	8722	30	60
WEING1	141278	2	28
WEING2	130883	2	28
WEING3	95677	2	28
WEING4	119337	2	28
WEING5	98796	2	28
WEING6	130623	2	28
WEING7	1095450	2	105
WEING8	624319	2	105
WEISH01	4554	5	30
WEISH02	4536	5	30
WEISH03	4115	5	30
WEISH04	4561	5	30
WEISH05	4514	5	30
WEISH06	5557	5	40
WEISH07	5567	5	40
WEISH08	5605	5	40
WEISH09	5246	5	40
WEISH10	6339	5	50
WEISH11	5643	5	50
WEISH12	6339	5	50
WEISH13	6159	5	50
WEISH14	6954	5	60
WEISH15	7486	5	60
WEISH16	7289	5	60
WEISH17	8633	5	60
WEISH18	9580	5	70
WEISH19	7698	5	70
WEISH20	9450	5	70
WEISH21	9074	5	70
WEISH22	8947	5	70
WEISH23	8344	5	80
WEISH24	10220	5	80
WEISH25	9939	5	80
WEISH26	9584	5	90
WEISH27	9819	5	90
WEISH28	9492	5	90
WEISH29	9410	5	90
WEISH30	11191	5	90
PB1	3090	4	27
PB2	3186	4	34
PB4	95168	2	29
PB5	2139	10	20
PB6	776	30	40
PB7	1035	30	37
HP1	3418	4	28
HP2	3186	4	35

ง.1 การทดลองปัญหาการติดในจุดสุดสัมพัทธ์

ในหัวข้อนี้วิเคราะห์ผลกระทบจากปัญหาถุงเป้แบบหลายมิติที่เกิดกับ BPSO โดยตาราง ง.2 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่สำหรับตารางนี้ได้ใช้จำนวนรอบการค้นหา 100 รอบ และ 1000 รอบ โดยตาราง ง.2 แสดงค่า MBF และ SR

จากตาราง ง.2 โดยภาพรวมแล้ว BPSO ที่มีจำนวนรอบการค้นหาคำตอบมาก (1000 รอบ) จะได้รับผลลัพธ์การค้นหาดีกว่า BPSO ที่มีจำนวนรอบการค้นหาคำตอบน้อย (100 รอบ) แต่ในบางปัญหาจำนวนรอบการค้นหาคำตอบที่น้อยกว่าก็ได้ผลลัพธ์ที่ดีกว่าจำนวนรอบการค้นหาคำตอบที่มาก เช่น WEING3, WEING7, WEISH11 และ WEISH26 ที่เป็นเช่นนี้เพราะปัญหาถุงเป้แบบหลายมิติเป็นปัญหามัลติโมดอล ซึ่งในบางครั้งการค้นหาคำตอบประสบปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น ซึ่งการเพิ่มจำนวนรอบการค้นหาคำตอบให้มากขึ้นก็ไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้

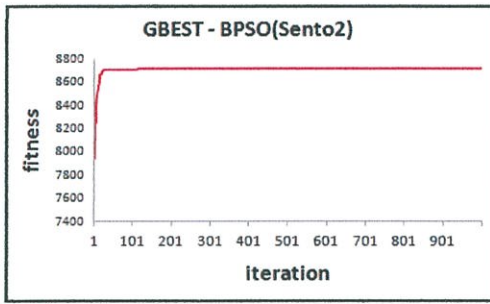
เพื่อแสดงให้เห็นผลกระทบจากปัญหาการติดในจุดสุดสัมพัทธ์ จึงได้ทำการทดลองโดยการค้นหาคำตอบด้วย BPSO กับปัญหา SENTO โดยใช้พารามิเตอร์ตามตารางที่ 5.2 แต่กราฟ ง.1 และ ง.3 ได้กำหนดจำนวนรอบการค้นหาคำตอบ 1000 รอบ ส่วนกราฟ ง.2 และ ง.4 ได้กำหนดจำนวนรอบการค้นหาคำตอบ 200 รอบ โดยกราฟ ง.1 และ ง.2 ได้แสดงค่าความเหมาะสมของ GBEST ของการค้นหาด้วย BPSO กับปัญหา SENTO โดยการค้นหานั้นจะดำเนินการเพียงครั้งเดียว ส่วนกราฟ ง.3 และ ง.4 แสดงค่าความเร็วของการค้นหาด้วย BPSO กับปัญหา SENTO โดยการค้นหานั้นจะดำเนินการเพียงครั้งเดียว

จากผลการทดลองสามารถสรุปได้ว่าการค้นหาคำตอบกับปัญหาถุงเป้แบบหลายมิติด้วย BPSO แม้จะเป็นปัญหาเดียวกัน แต่ในบางการค้นหาก็ประสบปัญหาการติดในจุดสุดสัมพัทธ์ ดังกราฟ ง.1 โดยหลังจากการค้นหาผ่านไประยะเวลาหนึ่ง คำตอบที่ได้จะไม่ดีขึ้นเลยและไม่ว่าเวลาจะผ่านไปนานเท่าใด คำตอบที่ได้ก็ไม่มีแนวโน้มว่าจะดีขึ้น แต่ในบางครั้งก็ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ซึ่งจะทำให้พบคำตอบที่ดีที่สุดดังกราฟ ง.2

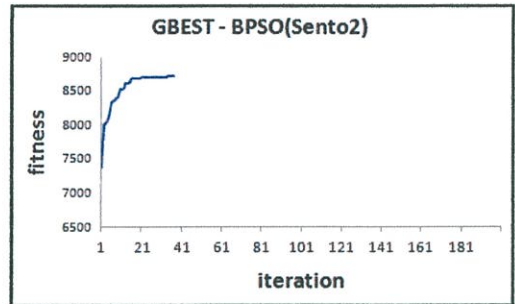
สำหรับกราฟ ง.3 และ ง.4 แสดงให้เห็นว่า ช่วงแรกของการค้นหาความเร็วของอนุภาคจะมีค่ามากและเมื่อเวลาผ่านไปความเร็วของอนุภาคจะมีค่าลดลง โดยเมื่อ BPSO เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น ความเร็วของอนุภาคจะมีแนวโน้มลดลงไปเรื่อยๆ โดยสรุปแล้วผลกระทบจากปัญหาการติดในจุดสุดสัมพัทธ์ของปัญหาถุงเป้หลายมิติให้ผลลัพธ์เหมือนกับฟังก์ชันมาตรฐาน

ตารางที่ ง.2 ผลการค้นหาคำตอบของปัญหาถุงเป้แบบหลายมิติด้วย BPSO

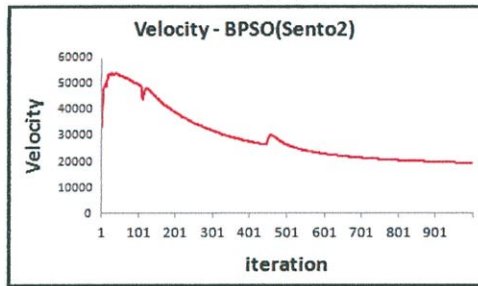
Algorithm problem	BPSO (100)		BPSO (1000)	
	MBF	SR	MBF	SR
SENTO1	7768.13	76	7768.72	79
SENTO2	8715.93	37	8715.57	37
WEING1	141278	100	141278	100
WEING2	130883	100	130883	100
WEING3	95629.4	89	95634.1	86
WEING4	119337	100	119337	100
WEING5	98796	100	98796	100
WEING6	130600	94	130607	96
WEING7	1095220	8	1095330	7
WEING8	620913	9	621222	17
WEISH01	4554	100	4554	100
WEISH02	4535.05	81	4535.3	86
WEISH03	4115	100	4115	100
WEISH04	4561	100	4561	100
WEISH05	4514	100	4514	100
WEISH06	5553.03	76	5554.78	86
WEISH07	5567	100	5567	100
WEISH08	5604.96	98	5604.98	99
WEISH09	5246	100	5246	100
WEISH10	6338.99	99	6339	100
WEISH11	5642.96	99	5642.88	97
WEISH12	6339	100	6339	100
WEISH13	6159	100	6159	100
WEISH14	6954	100	6954	100
WEISH15	7486	100	7486	100
WEISH16	7288.64	93	7288.72	94
WEISH17	8632.89	99	8633	100
WEISH18	9577.57	79	9579.08	88
WEISH19	7696.7	90	7697.22	94
WEISH20	9449	95	9449.6	98
WEISH21	9070.85	90	9074	100
WEISH22	8938.36	52	8938.54	53
WEISH23	8342.47	82	8343.01	89
WEISH24	10219.7	97	10219.2	96
WEISH25	9934.58	71	9933.58	68
WEISH26	9576.84	78	9575.8	75
WEISH27	9818.61	99	9819	100
WEISH28	9492	100	9492	100
WEISH29	9407.84	93	9410	100
WEISH30	11189.8	76	11190.1	78
PB1	3084.03	72	3082.04	64
PB2	3176.84	70	3176.37	63
PB4	95164.1	99	95168	100
PB5	2133.44	71	2132.75	75
PB6	775.16	94	775.72	98
PB7	1032.54	74	1033.12	82
HP1	3409.88	60	3412.29	70
HP2	3176.73	75	3178.31	74
ผลรวม		4075		4149



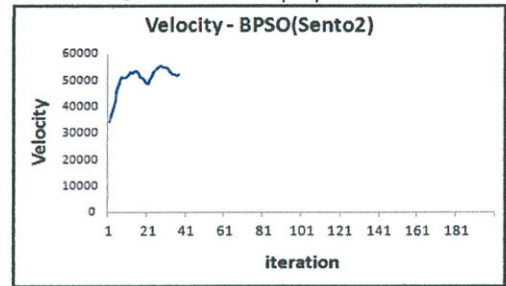
รูปที่ ๓.1 ค่าความเหมาะสมของ BPSO ที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์



รูปที่ ๓.2 ค่าความเหมาะสมของ BPSO ที่ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์



รูปที่ ๓.3 ความเร็วของ BPSO ที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์



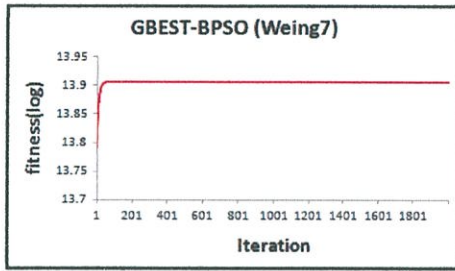
รูปที่ ๓.4 ความเร็วของ BPSO ที่ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์

๓.2 การทดลองการประยุกต์ใช้การกลายพันธุ์

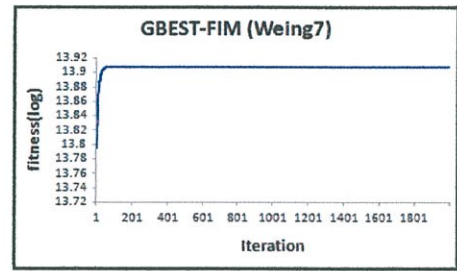
ในหัวข้อนี้พิจารณาถึงความสามารถในการค้นหาคำตอบด้วย BPSO, MXUX ซึ่งเป็นตัวแทนของการกลายพันธุ์แบบ CM และ FIM ซึ่งเป็นตัวแทนของการกลายพันธุ์แบบ NCM เพื่อวิเคราะห์ผลกระทบจากการกลายพันธุ์กับ BPSO ที่ดำเนินการแก้ปัญหาหุ่นยนต์แบบหลายมิติ

๓.2.1 การพิจารณาการกลายพันธุ์แบบ NCM

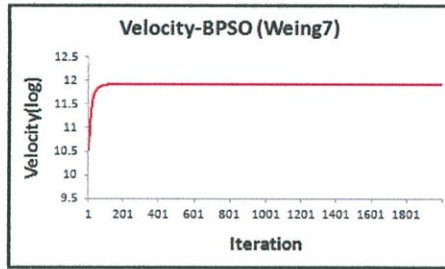
หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย BPSO และ FIM กับปัญหา WEING7 โดยใช้พารามิเตอร์ตามตาราง 5.2 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟ ๓.5 และ ๓.6 ได้แสดงค่า LMFG ของ BPSO และ FIM ตามลำดับ ส่วนกราฟ ๓.7 และ ๓.8 แสดงค่าลอการิทึมของ AVGV ของการค้นหาคำตอบด้วย BPSO และ FIM ตามลำดับ โดยกราฟ ๓.9 ได้แสดงค่า LD (t) ของการค้นหาคำตอบด้วย BPSO และ FIM โดย LD (t) คือค่าลอการิทึมของค่าเฉลี่ยของความหลากหลายของประชากรในแต่ละรอบการค้นหาคำตอบ สำหรับตารางการทดลอง ๓.3 คือการทดลองเปรียบเทียบผลลัพธ์ BPSO กับ FIM โดยใช้พารามิเตอร์ตามตารางที่ 5.2 โดยตาราง ๓.3 แสดงค่า MBF และ SR



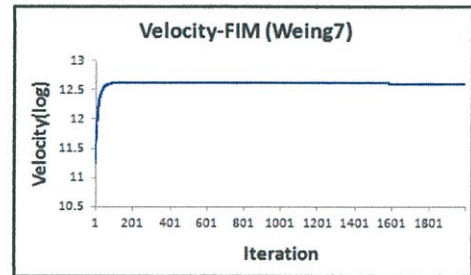
รูปที่ ๓.5 LMFG ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย BPSO



รูปที่ ๓.6 LMFG ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย FIM



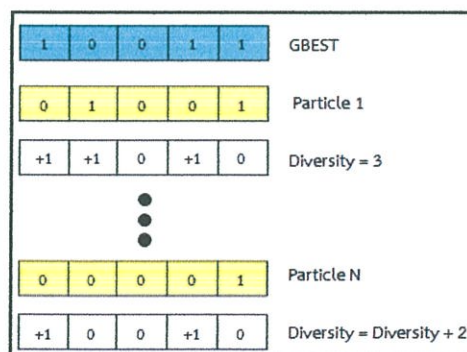
รูปที่ ๓.7 ค่าลอการิทึมของ AVGV ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย BPSO



รูปที่ ๓.8 ค่าลอการิทึมของ AVGV ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย FIM

จากกราฟ ๓.5 และ ๓.6 กราฟทั้งสองเกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยสังเกตจากเส้นกราฟมีค่าคงที่ เมื่อพิจารณากราฟ ๓.7 และ ๓.8 จะพบว่าความเร็วของ FIM มีค่ามากกว่า BPSO จึงทำให้อนุภาคกระจายตัวมากกว่าทำให้สามารถหลีกเลี่ยงหรือลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า BPSO ดังนั้นปัญหาถุงเป้แบบหลายมิตินั้นจึงให้ผลลัพธ์เช่นเดียวกับฟังก์ชันมาตรฐาน

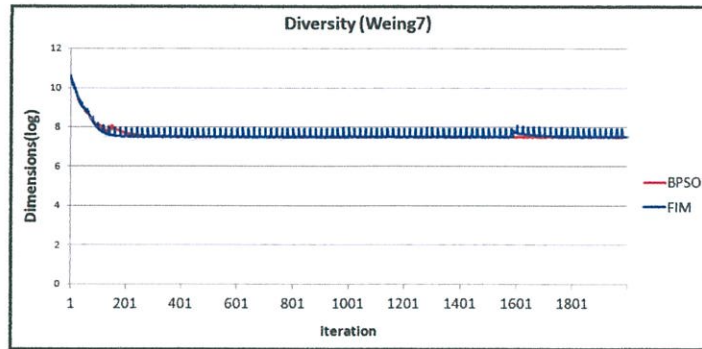
เนื่องจากปัญหาถุงเป้แบบหลายมิติไม่มีการคำนวณความหลากหลายของประชากร ดังนั้นงานวิจัยฉบับได้เสนอให้มีการคำนวณความหลากหลายโดยใช้วิธีการเปรียบเทียบความแตกต่างระหว่างตำแหน่งที่ดีที่สุดที่อนุภาคเคยค้นพบหรือ GBEST กับตำแหน่งของอนุภาคทั้งหมด ถ้าหากมีมิตินอนุภาคมีบิตที่แตกต่างจาก GBEST ค่าความหลากหลายก็จะเพิ่มขึ้นดังรูป ๓.9 แต่ถ้าหากมิติที่อยู่บนอนุภาคมีบิตที่เหมือนกับ GBEST ค่าความหลากหลายจะไม่เพิ่มขึ้นดังรูป ๓.9 เมื่อดำเนินการเช่นนี้จนครบทุกอนุภาค ผลลัพธ์ที่ได้ก็จะเป็นความหลากหลายของประชากรในรอบการค้นหาคำตอบนั้น



รูปที่ ๓.9 ตัวอย่างการคำนวณความหลากหลายของประชากรของ BPSO

ตารางที่ ง.3 ผลการค้นหาคำตอบของปัญหาถ่วงเบ้หลายมิติด้วย PSO และ FIM

Algorithm problem	BPSO		FIM	
	MBF	SR	MBF	SR
Sento1	7767.82	75	7769.11	83
Sento2	8716.29	37	8715.02	34
Weing1	141278	100	141278	100
Weing2	130883	100	130883	100
Weing3	95643.7	91	95620.7	88
Weing4	119337	100	119337	100
Weing5	98796	100	98758.2	99
Weing6	130611	97	130604	95
Weing7	1095330	9	1095340	16
Weing8	621019	13	621321	17
Weish01	4554	100	4554	100
Weish02	4535.3	86	4535.35	87
Weish03	4115	100	4115	100
Weish04	4561	100	4561	100
Weish05	4514	100	4514	100
Weish06	5553.24	78	5554.58	86
Weish07	5567	100	5567	100
Weish08	5604.94	97	5604.98	99
Weish09	5246	100	5246	100
Weish10	6339	100	6339	100
Weish11	5643	100	5642.96	99
Weish12	6339	100	6339	100
Weish13	6159	100	6159	100
Weish14	6954	100	6954	100
Weish15	7486	100	7486	100
Weish16	7288.88	94	7288.9	95
Weish17	8633	100	8633	100
Weish18	9578.35	87	9578.96	91
Weish19	7696.83	91	7697.09	93
Weish20	9449.6	98	9449.8	99
Weish21	9073.52	98	9072.8	95
Weish22	8938.18	51	8937.79	50
Weish23	8343.67	89	8343.34	89
Weish24	10220	100	10219.5	98
Weish25	9934.66	73	9935.27	77
Weish26	9574.93	73	9576.11	77
Weish27	9819	100	9819	100
Weish28	9492	100	9492	100
Weish29	9410	100	9409.68	99
Weish30	11190.2	81	11190.1	77
PB1	3083.39	72	3083.68	64
PB2	3178.33	69	3175.68	71
PB4	95168	100	95168	100
PB5	2133.53	70	2133.73	69
PB6	775.58	97	775.58	97
PB7	1033.67	85	1033.71	87
HP1	3410.74	65	3411.06	64
HP2	3178.79	75	3180.47	80
ผลรวม		4151		4175

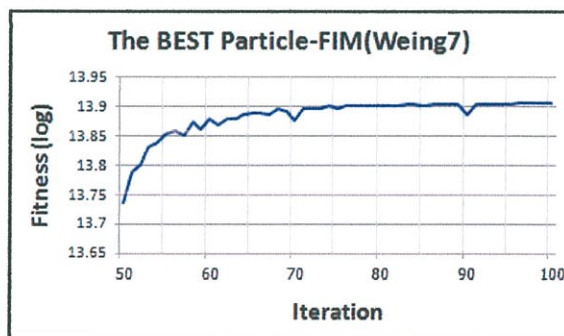


รูปที่ ๑.10 LD(t) ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย PSO และ FIM

จากกราฟ ๑.10 แสดงให้เห็นว่าทั้ง BPSO และ FIM จะมีความหลากหลายของประชากรลดลงแล้วคงที่เมื่ออนุภาคติดในจุดสุดสัมพัทธ์ สำหรับ BPSO เมื่ออนุภาคติดในจุดสุดสัมพัทธ์ความหลากหลายของประชากรจะมีคงที่ แต่สำหรับ FIM ความหลากหลายของประชากรจะมีค่าเพิ่มขึ้นเป็นช่วงๆ ส่งผลให้ FIM มีขอบเขตการค้นหาที่กว้างกว่า BPSO จึงมีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ได้มากกว่า จากตาราง ๑.3 แสดงให้เห็นว่า FIM ได้ผลลัพธ์โดยรวมดีกว่า BPSO เพราะ FIM มีความหลากหลายของประชากรมากกว่า BPSO จึงสามารถลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า BPSO โดยสรุปแล้วปัญหาถุงเป้แบบหลายมิตินั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ FIM มีความหลากหลายมากกว่าและได้รับคำตอบจากการค้นหาที่ดีกว่า BPSO

๑.2.2 การพิจารณาการเพิ่มประสิทธิภาพการค้นหาของ PSO ด้วย NCM

หัวข้อนี้ได้ทดลองการค้นหาคำตอบด้วย FIM กับปัญหา WEING7 โดยกราฟ ๑.11 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหาคำตอบ 200 รอบ กำหนดรอบการกลายพันธุ์เท่ากับ 10 และจำนวนครั้งในการทดลอง 1 ครั้งเท่านั้น โดยกราฟแสดงค่าของ LBP ของการค้นหาด้วย FIM กับปัญหา WEING7 ในรอบการค้นหาช่วง 50 ถึง 100 รอบ



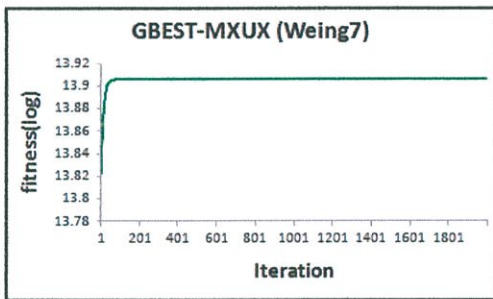
รูปที่ ๑.11 FIM ปรับปรุง BPSO ในช่วงรอบการค้นหา 50 ถึง 100

จากรูป ๑.11 การกลายพันธุ์ถูกกำหนดให้ดำเนินการทุก 10 รอบของการค้นหา โดยการกลายพันธุ์ทำให้เกิดการกระจายตัวของอนุภาคออกจากตำแหน่งของ GBEST ไปยังตำแหน่งที่แย่กว่าเดิมสังเกตได้จากเส้นกราฟเมื่อผ่านการกลายพันธุ์เส้นกราฟจะมีค่าลดลง หลังจากนั้นอนุภาคจะทำการลู่เข้าหากันใหม่ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้คำตอบนั้นดีขึ้นสังเกตได้

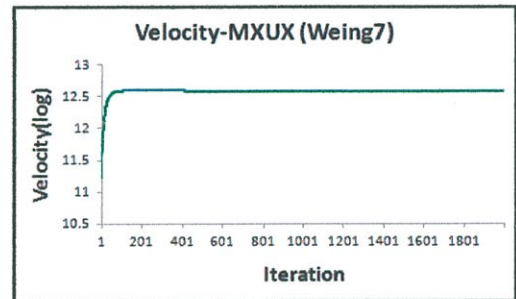
จากเส้นกราฟมีค่าเพิ่มขึ้นมากกว่าก่อนการกลายพันธุ์ จากผลการทดลองสรุปว่าการกลายพันธุ์ทำให้อุณหภูมิกระจายตัวออกจากกันแล้ว การปรับปรุงค่าตอบเกิดจากการที่อนุภาคคู่แข่งหาหาคัน ซึ่งผลกระทบจากการปรับปรุงค่าตอบ NCM ที่ใช้กับปัญหาถุงเป้หลายมิติเหมือนกับผลกระทบจากการปรับปรุงค่าตอบของ NCM ที่ใช้กับฟังก์ชันมาตรฐาน

ง.2.3 การพิจารณาการกลายพันธุ์แบบ CM

หัวข้อนี้ได้ทดลองค้นหาค่าตอบด้วย BPSO และ MXUX กับปัญหา WEING7 โดยกราฟ ง.12, ง.13, ง.15 ใช้พารามิเตอร์ตามตาราง 5.2 แต่สำหรับจำนวนรอบการค้นหาค่าตอบ 2000 รอบ โดยกราฟ ง.12 แสดงค่า LMFG ของ MXUX ส่วนกราฟ ง.13 แสดงค่าลอการิทึมของ AVGV ของการค้นหาค่าด้วย MXUX กราฟ ง.15 แสดงค่า LD (t) ของการค้นหาค่าด้วย BPSO และ FIM สำหรับตารางการทดลอง ง.4 คือผลลัพธ์การค้นหาค่าตอบด้วย BPSO กับ MXUX โดยใช้พารามิเตอร์ตามตาราง 5.2 โดยตาราง ง.4 แสดงค่า MBF และ SR



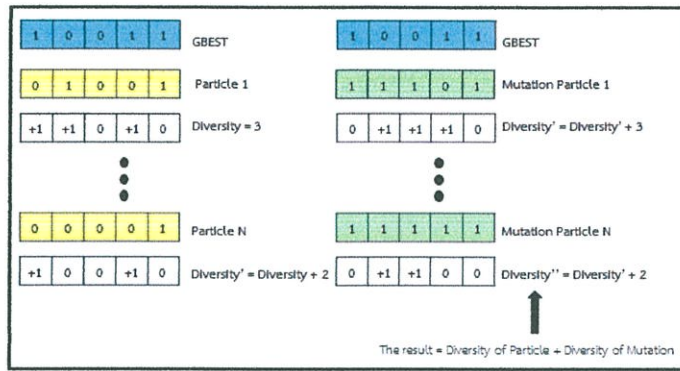
รูปที่ ง.12 LMFG ของปัญหา WEING7 ที่ค้นหาค่าตอบด้วย MXUX



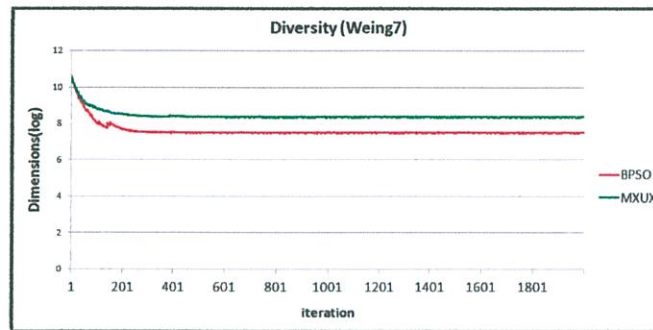
รูปที่ ง.13 ค่าลอการิทึม AVGV ของปัญหา WEING7 ที่ค้นหาค่าตอบด้วย MXUX

กราฟ ง.12 เกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยสังเกตจากเส้นกราฟมีค่าคงที่เมื่อพิจารณากราฟ ง.13 และ ง.7 พบว่าความเร็วของ MXUX มีค่ามากกว่า BPSO จึงทำให้อุณหภูมิกระจายตัวมากกว่าทำให้สามารถหลุดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า BPSO

จากหัวข้อ ง.2.1 ได้อธิบายเกี่ยวกับการคำนวณความหลากหลายของประชากรในกรณี BPSO และ FIM ซึ่ง MXUX ก็มีขั้นตอนการคำนวณความหลากหลายของประชากรเหมือนกับ BPSO และ FIM เพียงแต่กรณี MXUX ต้องมีขั้นตอนการคำนวณความหลากหลายของประชากรเพิ่มเติมคือต้องเปรียบเทียบความแตกต่างของ GBEST กับตำแหน่งที่เกิดจากการดำเนินการกลายพันธุ์ดังรูป ง.14 โดยการคำนวณความหลากหลายของประชากรใช้วิธีการคำนวณเหมือนกับกรณี BPSO และ FIM



รูปที่ ง.14 ตัวอย่างการคำนวณความหลากหลายของประชากรของ MXUX



รูปที่ ง.15 LD(t) ของปัญหา WEING7 ด้วย PSO และ MXUX

จากกราฟ ง.15 จะเห็นว่าเส้นกราฟของ BPSO ในช่วงคงที่นั้นมีค่าน้อยกว่าเส้นกราฟของ MXUX ในช่วงคงที่หมายความว่าในขณะที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ความหลากหลายประชากรของ BPSO จะมีค่าน้อยกว่าความหลากหลายประชากรของ MXUX ส่งผลให้ MXUX มีขอบเขตการค้นหาที่กว้างกว่าและมีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ได้มากกว่า BPSO

จากตาราง ง.4 พบว่า MXUX ได้ผลลัพธ์โดยรวมดีกว่า BPSO เพราะ MXUX สามารถเพิ่มความหลากหลายให้กับประชากรจึงสามารถลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า BPSO ดังนั้นการกลายพันธุ์แบบ CM สามารถช่วยปรับปรุงคำตอบและเพิ่มประสิทธิภาพการค้นหาของ BPSO โดยสรุปแล้วปัญหาถุงเป้แบบหลายมิตินี้ให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MXUX มีความหลากหลายมากกว่าและได้รับคำตอบจากการค้นหาที่ดีกว่า BPSO

ตารางที่ ง.4 ผลการค้นหาคำตอบของปัญหาถุงเป้หลายมิติด้วย BPSO และ MXUX

Algorithm	BPSO		MXUX	
	MBF	SR	MBF	SR
Sento1	7767.82	75	7769.59	84
Sento2	8716.29	37	8716.29	42
Weing1	141278	100	141278	100
Weing2	130883	100	130883	100
Weing3	95643.7	91	95622.2	91
Weing4	119337	100	119337	100
Weing5	98796	100	98796	100
Weing6	130611	97	130623	100
Weing7	1095330	9	1095350	10
Weing8	621019	13	621403	22
Weish01	4554	100	4554	100
Weish02	4535.3	86	4535	80
Weish03	4115	100	4115	100
Weish04	4561	100	4561	100
Weish05	4514	100	4514	100
Weish06	5553.24	78	5554.19	83
Weish07	5567	100	5567	100
Weish08	5604.94	97	5605	100
Weish09	5246	100	5246	100
Weish10	6339	100	6339	100
Weish11	5643	100	5643	100
Weish12	6339	100	6339	100
Weish13	6159	100	6159	100
Weish14	6954	100	6954	100
Weish15	7486	100	7486	100
Weish16	7288.88	94	7289	100
Weish17	8633	100	8633	100
Weish18	9578.35	87	9579.72	96
Weish19	7696.83	91	7697.87	99
Weish20	9449.6	98	9450	100
Weish21	9073.52	98	9074	100
Weish22	8938.18	51	8939.44	58
Weish23	8343.67	89	8343.85	95
Weish24	10220	100	10219.8	99
Weish25	9934.66	73	9935.55	79
Weish26	9574.93	73	9573.58	69
Weish27	9819	100	9819	100
Weish28	9492	100	9492	100
Weish29	9410	100	9409.68	99
Weish30	11190.2	81	11190.4	85
PB1	3083.39	72	3085.88	74
PB2	3178.33	69	3182.56	84
PB4	95168	100	95168	100
PB5	2133.53	70	2136.28	84
PB6	775.58	97	775.16	94
PB7	1033.67	85	1033.6	85
HP1	3410.74	65	3412.7	66
HP2	3178.79	75	3183.04	91
ผลรวม		4151		4269

ง.3 การทดลองปัญหาจากการกลายพันธุ์แบบ NCM

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากปัญหาที่เกิดจากการกลายพันธุ์แบบ NCM กับปัญหาถุงเป้หลายมิติ

ง.3.1 การพิจารณาปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

ตารางที่ ง.5 ผลการทดลองปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบสำหรับ MKP

Algorithm problem	BPSO		FIM (Over mutation)	
	MBF	SR	MBF	SR
Sento1	7767.82	75	7762.75	57
Sento2	8716.29	37	8711.48	16
Weing1	141278	100	141278	100
Weing2	130883	100	130883	100
Weing3	95643.7	91	95677	100
Weing4	119337	100	119337	100
Weing5	98796	100	98796	100
Weing6	130611	97	130619	99
Weing7	1095330	9	1095350	16
Weing8	621019	13	603599	0
Weish01	4554	100	4554	100
Weish02	4535.3	86	4535.2	84
Weish03	4115	100	4115	100
Weish04	4561	100	4561	100
Weish05	4514	100	4514	100
Weish06	5553.24	78	5552.29	68
Weish07	5567	100	5567	100
Weish08	5604.94	97	5604.98	99
Weish09	5246	100	5246	100
Weish10	6339	100	6338.73	99
Weish11	5643	100	5642.72	93
Weish12	6339	100	6338.99	99
Weish13	6159	100	6159	100
Weish14	6954	100	6954	100
Weish15	7486	100	7486	100
Weish16	7288.88	94	7288.74	98
Weish17	8633	100	8633	100
Weish18	9578.35	87	9577.36	81
Weish19	7696.83	91	7697.22	94
Weish20	9449.6	98	9450	100
Weish21	9073.52	98	9074	100
Weish22	8938.18	51	8931.88	16
Weish23	8343.67	89	8340.64	87
Weish24	10220	100	10219	95
Weish25	9934.66	73	9932.12	60
Weish26	9574.93	73	9579.53	86
Weish27	9819	100	9819	100
Weish28	9492	100	9492	100
Weish29	9410	100	9404.56	83
Weish30	11190.2	81	11189.9	79
PB1	3083.39	72	3083.68	65
PB2	3178.33	69	3172.68	59
PB4	95168	100	95164.1	99
PB5	2133.53	70	2133.24	72
PB6	775.58	97	775.3	95
PB7	1033.67	85	1033.64	86
HP1	3410.74	65	3411.21	64
HP2	3178.79	75	3177.65	70
ผลรวม		4151		4019

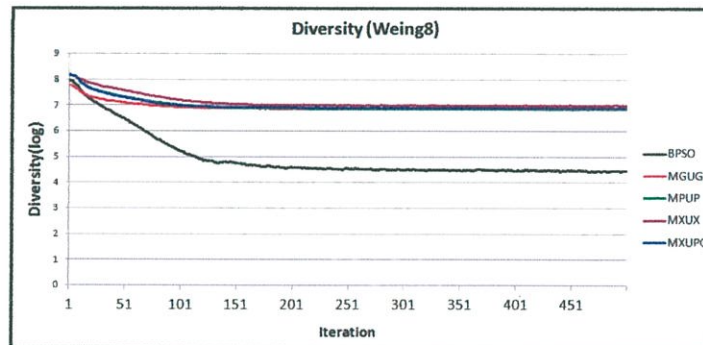
จากตารางผลการทดลอง ง.5 ได้ค้นหาคำตอบด้วย BPSO และ FIM กับปัญหาถุงเป้หลายมิติ โดยใช้พารามิเตอร์ตามตาราง 5.2 แต่สำหรับการทดลองนี้รอบของการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 1 รอบและความน่าจะเป็นในการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 0.30 โดยตาราง ง.5 แสดงค่า MBF และ SR จากผลการทดลองตารางนี้แสดงให้เห็นว่า ผลลัพธ์ที่ได้นั้นเหมือนกรณีของฟังก์ชันมาตรฐาน คือการกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป (1 รอบ และ 30 เปอร์เซ็นต์) ผลลัพธ์จากการค้นหา FIM โดยรวมแล้วแยกว่า BPSO

ง.4 การทดลองปัญหาและแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากปัญหาที่เกิดจากการกลายพันธุ์แบบ CM และแนวทางการแก้ปัญหาที่เกิดจากการกลายพันธุ์แบบ CM กับปัญหาถุงเป้หลายมิติ

ง.4.1 การพิจารณาความหลากหลายของประชากรของการกลายพันธุ์แบบ CM

สำหรับกราฟ ง.16 คือการทดลองค้นหาคำตอบด้วย BPSO, MXUX, MPUP, MGUG และ MXUPG กับปัญหา WEING8 โดยใช้พารามิเตอร์ตามตาราง 5.2 แต่จำนวนรอบการค้นหาคำตอบ 500 รอบ โดยกราฟแสดงค่า LD(t) ของการค้นหา

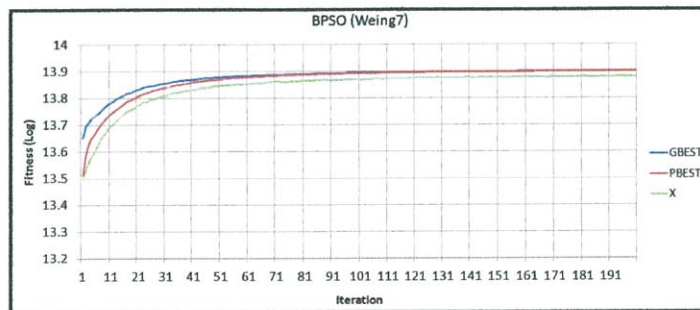


รูปที่ ง.16 LD(t) ของ BPSO, MGUG, MPUP, MXUX และ MXUPG ของปัญหา WEING8

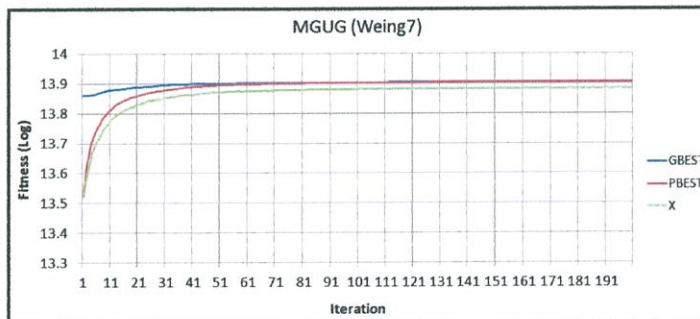
จากกราฟ ง.16 ความหลากหลายของประชากรในช่วงเริ่มต้นของการค้นหาของ BPSO, MGUG, MPUP, MXUX และ MXUPG จะมีค่ามาก เมื่อเวลาผ่านไปความหลากหลายของประชากรจะมีค่าลดลง และเมื่ออนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ความหลากหลายของประชากรจะอยู่ในระดับคงที่ ในช่วงที่ระดับคงที่เส้นกราฟของ MGUG, MPUP, MXUX และ MXUPG จะมีระดับใกล้เคียงกันและมีระดับที่มากกว่า BPSO หมายความว่า ในช่วงที่อนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ความหลากหลายของประชากรของ MGUG, MPUP, MXUX และ MXUPG จะมีค่าใกล้เคียงกันและมีค่ามากกว่าความหลากหลายของประชากรของ BPSO โดยสรุปแล้วปัญหาถุงเป้แบบหลายมิตินั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MGUG, MPUP, MXUX และ MXUPG มีความหลากหลายมากกว่า BPSO

ง.4.2 การพิจารณาปัญหาการลู่เร็วเกินไปและแนวทางแก้ปัญหาการลู่เร็วเกินไป

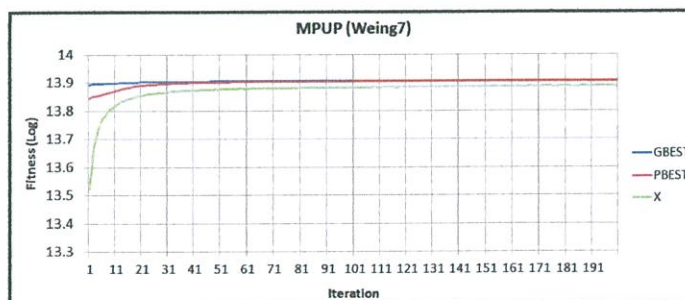
กราฟ ง.17 ถึง ง.21 ได้ทดลองค้นหาค่าตอบด้วย BPSO, MXUX, MPUP, MGUG และ MXUPG กับปัญหา WEING7 โดยใช้พารามิเตอร์ตามตาราง 5.2 แต่ได้กำหนดจำนวนรอบการค้นหาค่าตอบ 200 รอบ สำหรับ MGUG, MPUP, MXUX และ MXUPG จำนวนอนุภาคถูกกำหนดให้เท่ากับ 100 อนุภาค จำนวนรอบของการกลายพันธุ์ 9 รอบ และความน่าจะเป็นในการกลายพันธุ์ได้กำหนดค่าเท่ากับ 0.30 โดยกราฟแสดงค่า LMF_X, LMF_P และ LMF_G ของการค้นหาด้วย BPSO, MGUG, MPUP, MXUX และ MXUPG ส่วนตาราง ง.6 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่ได้กำหนดจำนวนอนุภาคให้เท่ากับ 100 อนุภาค จำนวนรอบของการกลายพันธุ์ 9 รอบ และความน่าจะเป็นในการกลายพันธุ์เท่ากับ 0.30 ส่วนตาราง ง.7 ได้ใช้พารามิเตอร์ตามตาราง 5.2 โดยตาราง ง.6 และ ง.7 แสดงค่า MBF และ SR



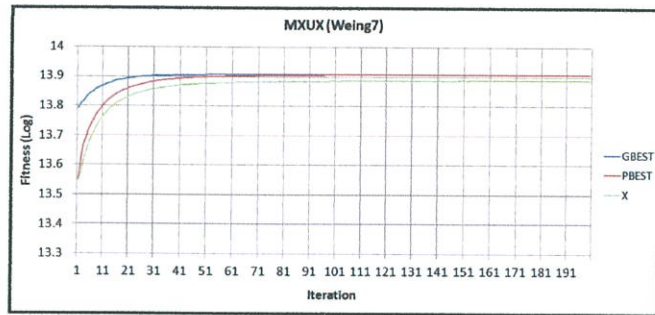
รูปที่ ง.17 LMF_X, LMF_P และ LMF_G ของ BPSO ของปัญหา WEING7



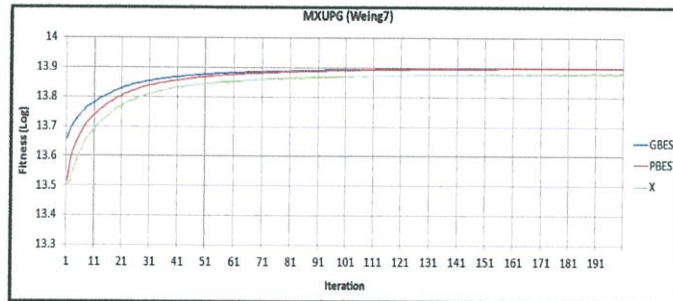
รูปที่ ง.18 LMF_X, LMF_P และ LMF_G ของ MGUG ของปัญหา WEING7



รูปที่ ง.19 LMF_X, LMF_P และ LMF_G ของ MPUP ของปัญหา WEING7



รูปที่ ง.20 LMFx, LMFP และ LMFG ของ MXUX ของปัญหา WEING7



รูปที่ ง.21 LMFx, LMFP และ LMFG ของ MXUPG ของปัญหา WEING7

จากกราฟ ง.17 ถึง ง.21 ในช่วงแรกของการค้นหาเส้นกราฟของ LMFx, LMFP, LMFG ของ BPSO, MGUG, MPUP, MXUX, MXUPG จะมีค่าเพิ่มขึ้นเรื่อยๆ เพราะอนุภาคพยายามลู่เข้าหากันตามหลักการทำงานของ BPSO และเมื่ออนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ เส้นกราฟทั้ง 3 จะเข้าสู่สถานะคงที่จนสิ้นสุดการค้นหา เส้นกราฟของ LMFx ของ MGUG, MPUP, MXUX รอบของการค้นหาเข้าสู่สถานะคงที่ประมาณ 80, 70 และ 80 รอบตามลำดับ ในขณะที่เส้นกราฟ LMFx ของ BPSO และ MXUPG รอบของการค้นหาเข้าสู่สถานะคงที่ประมาณ 110 รอบ และ 110 ตามลำดับ ซึ่ง MGUG, MPUP, MXUX เข้าสู่สภาพคงที่เร็วกว่า BPSO และ MXUPG หมายความว่า MGUG, MPUP, MXUX มีความเร็วในการลู่เข้าหากันเร็วกว่า BPSO และ MXUPG ซึ่งทำให้ MGUG, MPUP, MXUX เกิดปัญหาการลู่เร็วเกินไป โดยสรุปแล้วปัญหาถุงเป้แบบหลายมิตินี้ให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MGUG, MPUP, MXUX มีความเร็วในการลู่เข้าหากันมากกว่า BPSO และ MXUPG

ตารางที่ ๖.6 ผลการทดลองปัญหาการลู่เร็วเกินไปสำหรับ MKP

Algorithm	PSO		MXUX (Over mutation)	
	MBF	SR	MBF	SR
Sento1	7767.82	75	7766.34	62
Sento2	8716.29	37	8714.76	26
Weing1	141278	100	141278	100
Weing2	130883	100	130883	100
Weing3	95643.7	91	95166	27
Weing4	119337	100	119337	100
Weing5	98796	100	98720.4	98
Weing6	130611	97	130623	100
Weing7	1095330	9	1095190	3
Weing8	621019	13	620923	16
Weish01	4554	100	4554	100
Weish02	4535.3	86	4534.8	76
Weish03	4115	100	4115	100
Weish04	4561	100	4561	100
Weish05	4514	100	4514	100
Weish06	5553.24	78	5554.66	87
Weish07	5567	100	5567	100
Weish08	5604.94	97	5605	100
Weish09	5246	100	5246	100
Weish10	6339	100	6339	100
Weish11	5643	100	5643	100
Weish12	6339	100	6339	100
Weish13	6159	100	6159	100
Weish14	6954	100	6954	100
Weish15	7486	100	7486	100
Weish16	7288.88	94	7288.98	99
Weish17	8633	100	8633	100
Weish18	9578.35	87	9578.87	91
Weish19	7696.83	91	7697.09	93
Weish20	9449.6	98	9448.2	91
Weish21	9073.52	98	9074	100
Weish22	8938.18	51	8936.2	40
Weish23	8343.67	89	8343.97	99
Weish24	10220	100	10219.3	94
Weish25	9934.66	73	9933.84	68
Weish26	9574.93	73	9570.46	58
Weish27	9819	100	9819	100
Weish28	9492	100	9491.08	96
Weish29	9410	100	9402.8	77
Weish30	11190.2	81	11190.3	82
PB1	3083.39	72	3076.82	41
PB2	3178.33	69	3183.66	87
PB4	95168	100	95071	97
PB5	2133.53	70	2132.2	60
PB6	775.58	97	773.9	85
PB7	1033.67	85	1029.8	46
HP1	3410.74	65	3401.88	38
HP2	3178.79	75	3183.2	88
ผลรวม		4151		3925

ตารางที่ ง.7 ผลการค้นหาคำตอบด้วย BPSO, MGUG, MXUX, MPUP และ MXUPG กับ MKP

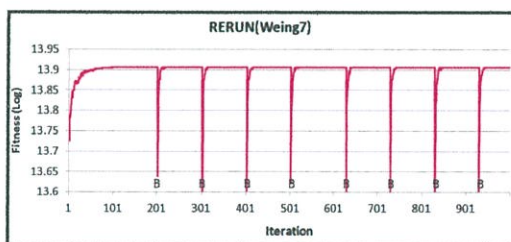
Algorithm	BPSO		MGUG		MPUP		MXUX		MXUPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
Sento1	7767.	75	7766.78	70	7768.91	84	7769.59	84	7771.72	98
Sento2	8716.	37	8715.1	29	8717.68	61	8716.29	42	8719.64	76
Weing1	1412	100	141278	100	141278	100	141278	100	141278	100
Weing2	1308	100	130883	100	130883	100	130883	100	130883	100
Weing3	9564	91	95605	71	95651.4	92	95622.2	91	95676.5	99
Weing4	1193	100	119337	100	119337	100	119337	100	119337	100
Weing5	9879	100	98796	100	98796	100	98796	100	98796	100
Weing6	1306	97	130607	96	130623	100	130623	100	130623	100
Weing7	1095	9	1095270	6	1095370	3	1095350	10	1095300	8
Weing8	6210	13	620859	14	620833	10	621403	22	624319	100
Weish01	4554	100	4554	100	4554	100	4554	100	4554	100
Weish02	4535.	86	4534.9	78	4535.4	88	4535	80	4536	100
Weish03	4115	100	4115	100	4115	100	4115	100	4115	100
Weish04	4561	100	4561	100	4561	100	4561	100	4561	100
Weish05	4514	100	4514	100	4514	100	4514	100	4514	100
Weish06	5553.	78	5553.6	80	5555.97	94	5554.19	83	5556.87	99
Weish07	5567	100	5567	100	5567	100	5567	100	5567	100
Weish08	5604.	97	5604.88	94	5605	100	5605	100	5605	100
Weish09	5246	100	5246	100	5246	100	5246	100	5246	100
Weish10	6339	100	6339	100	6339	100	6339	100	6339	100
Weish11	5643	100	5642.92	98	5643	100	5643	100	5643	100
Weish12	6339	100	6339	100	6339	100	6339	100	6339	100
Weish13	6159	100	6159	100	6159	100	6159	100	6159	100
Weish14	6954	100	6954	100	6954	100	6954	100	6954	100
Weish15	7486	100	7486	100	7486	100	7486	100	7486	100
Weish16	7288.	94	7288.58	79	7288.96	98	7289	100	7288.96	98
Weish17	8633	100	8633	100	8633	100	8633	100	8633	100
Weish18	9578.	87	9579.09	87	9580	100	9579.72	96	9580	100
Weish19	7696.	91	7694.62	74	7697.61	97	7697.87	99	7698	100
Weish20	9449.	98	9449.4	97	9450	100	9450	100	9450	100
Weish21	9073.	98	9074	100	9074	100	9074	100	9074	100
Weish22	8938.	51	8938.54	53	8939.08	56	8939.44	58	8943.58	81
Weish23	8343.	89	8343.52	84	8344	100	8343.85	95	8343.94	98
Weish24	1022	100	10219.8	99	10220	100	10219.8	99	10220	100
Weish25	9934.	73	9933.49	68	9935.32	77	9935.55	79	9938.01	93
Weish26	9574.	73	9560.8	35	9572.06	63	9573.58	69	9578.24	82
Weish27	9819	100	9819	100	9819	100	9819	100	9819	100
Weish28	9492	100	9492	100	9492	100	9492	100	9492	100
Weish29	9410	100	9409.68	99	9410	100	9409.68	99	9410	100
Weish30	1119	81	11190.8	96	11190.8	96	11190.4	85	11190.8	94
PB1	3083.	72	3083.62	71	3083.91	66	3085.88	74	3090	100
PB2	3178.	69	3168.69	59	3181.25	81	3182.56	84	3185.82	99
PB4	9516	100	95164.1	99	95168	100	95168	100	95168	100
PB5	2133.	70	2135.77	81	2134.75	75	2136.28	84	2139	100
PB6	775.5	97	775.58	97	776	100	775.16	94	776	100
PB7	1033.	85	1032.59	72	1034.23	91	1033.6	85	1034.91	99
HP1	3410.	65	3410.53	65	3411.18	70	3412.7	66	3418	100
HP2	3178.	75	3175.28	64	3183.29	85	3183.04	91	3186	100
ผลรวม		4151		4015		4287		4269		4624

จากตารางผลการทดลอง ง.6 ผลลัพธ์การค้นหาโดยรวมของ MXUX แย่กว่า BPSO เนื่องจาก MXUX เกิดปัญหาการลู่เร็วเกินไปมากกว่า BPSO จึงทำให้ผลลัพธ์การค้นหาคำตอบแย่กว่า BPSO จากตารางผลการทดลอง ง.7 ผลลัพธ์การค้นหาโดยรวมของ MPUP, MXUX ดีกว่า MGUG, BPSO และโดยรวม MXUPG ได้ผลลัพธ์ที่ดีที่สุด โดยสาเหตุที่ MXUPG สามารถหาคำตอบได้ดีที่สุดนั้นเป็นผลมาจากที่ MXUPG ไม่ทำให้ปัญหาการลู่เร็วเกินไปรุนแรงกว่า BPSO ปกติ จึงทำให้ผลลัพธ์การค้นหาคำตอบดีกว่า MXUX, MGUG, MPUP โดยสรุปแล้วปัญหาการลู่เร็วแบบหลายมิตินั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MXUPG ได้ผลลัพธ์ที่ดีที่สุดเมื่อเปรียบเทียบกับ MGUG, MPUP, MXUX และ BPSO และเมื่อกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป MXUX จะได้ผลลัพธ์การค้นหาแย่กว่า BPSO

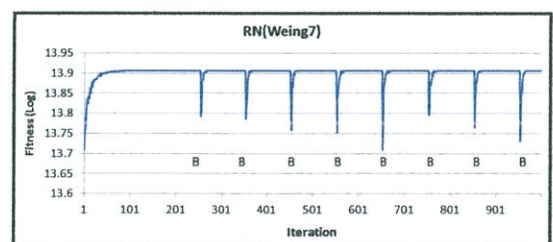
ง.5 การทดลองการประยุกต์ใช้รีรันและรีโพซิชั่น

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากรีรันและรีโพซิชั่นกับปัญหาการลู่เร็วหลายมิติ โดยการทดลองนี้ได้เลือกใช้ RN เป็นตัวแทนของรีโพซิชั่น เพราะทั้ง RN และ RERUN ไม่มีขั้นตอนการรีเซ็ตทั้ง GBEST และ PBEST

จากกราฟ ง.22 และ ง.23 คือ การทดลองรีโพซิชั่นและรีรันโดยใช้พารามิเตอร์ตามตาราง 5.2 แต่จำนวนรอบการค้นหาคำตอบ 1000 รอบ และจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้ง เพราะแต่ละรอบของการทดลองการรีเซ็ตเกิดขึ้นไม่พร้อมกัน ถ้าเป็นค่าเฉลี่ยจะเกิดการรวมค่าแล้วหักล้างกันทำให้ไม่พบผลของการรีเซ็ต โดยกราฟทั้งสองแสดงค่า LBP ของการค้นหาด้วยรีรันและรีโพซิชั่นกับปัญหา WEING7 ส่วนตาราง ง.8 คือ การทดลองรีโพซิชั่นและรีรันโดยทดสอบกับปัญหาการลู่เร็วหลายมิติโดยใช้พารามิเตอร์ตามตาราง 5.2 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหา 1000 รอบ โดยตารางแสดงค่า ARR ส่วนตาราง ง.9 คือการทดลองรีโพซิชั่นและรีรันโดยทดสอบกับปัญหาการลู่เร็วหลายมิติโดยใช้พารามิเตอร์ตามตาราง 5.2 โดยตาราง ง.9 แสดงค่า MBF และ SR



รูปที่ ง.22 LBP ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย RERUN



รูปที่ ง.23 LBP ของปัญหา WEING7 ที่ค้นหาคำตอบด้วย RN

ตารางที่ ง.8 ผลของ ARR ของปัญหาถูกแก้หลายมิติด้วย RERUN และ REPOSITION

Algorithm	RE-RUN	RE-POSITION (RN)
problem	ARR	ARR
Sento1	9	18.98
Sento2	8.95	18.78
Weing1	9	19
Weing2	9	19
Weing3	9.01	18.99
Weing4	9	19
Weing5	9	19
Weing6	9	19
Weing7	8.74	17.71
Weing8	8.99	18.7
Weish01	9	19
Weish02	9	19
Weish03	9	19
Weish04	9.02	19.02
Weish05	9.05	19.08
Weish06	9	18.97
Weish07	9	19
Weish08	9	19
Weish09	9	19
Weish10	9	19
Weish11	9	19
Weish12	9	19
Weish13	9	19
Weish14	9	19
Weish15	9	19
Weish16	8.98	18.9
Weish17	9	18.99
Weish18	8.98	18.81
Weish19	9	18.92
Weish20	8.98	18.91
Weish21	9	18.85
Weish22	9	18.89
Weish23	8.99	18.67
Weish24	8.97	18.72
Weish25	9	18.73
Weish26	8.99	18.47
Weish27	9	18.88
Weish28	9	18.79
Weish29	8.95	18.53
Weish30	8.97	18.39
PB1	9	18.99
PB2	8.98	18.92
PB4	9.02	19.02
PB5	9	19.01
PB6	9.02	19.04
PB7	8.98	18.98
HP1	9	18.97
HP2	8.98	18.94
ผลรวม	431.55	906.55

ตารางที่ 9.9 ผลการค้นหาคำตอบของปัญหาถุงเป้หลายมิติด้วย BPSO, RERUN และ REPOSITION

Algorithm problem	BPSO		RE-RUN		RE-POSITION (RN)	
	MBF	SR	MBF	SR	MBF	SR
Sento1	7767.82	75	7768.92	78	7769.62	83
Sento2	8716.29	37	8718.41	58	8718.53	59
Weing1	141278	100	141278	100	141278	100
Weing2	130883	100	130883	100	130883	100
Weing3	95643.7	91	95677	100	95677	100
Weing4	119337	100	119337	100	119337	100
Weing5	98796	100	98796	100	98796	100
Weing6	130611	97	130623	100	130623	100
Weing7	1095330	9	1095340	6	1095350	11
Weing8	621019	13	620801	6	621242	16
Weish01	4554	100	4554	100	4554	100
Weish02	4535.3	86	4535.1	82	4535.1	82
Weish03	4115	100	4115	100	4115	100
Weish04	4561	100	4561	100	4561	100
Weish05	4514	100	4514	100	4514	100
Weish06	5553.24	78	5555.01	87	5555.63	91
Weish07	5567	100	5567	100	5567	100
Weish08	5604.94	97	5605	100	5605	100
Weish09	5246	100	5246	100	5246	100
Weish10	6339	100	6339	100	6339	100
Weish11	5643	100	5643	100	5643	100
Weish12	6339	100	6339	100	6339	100
Weish13	6159	100	6159	100	6159	100
Weish14	6954	100	6954	100	6954	100
Weish15	7486	100	7486	100	7486	100
Weish16	7288.88	94	7289	100	7289	100
Weish17	8633	100	8633	100	8633	100
Weish18	9578.35	87	9579.7	98	9580	100
Weish19	7696.83	91	7697.22	94	7697.48	96
Weish20	9449.6	98	9450	100	9450	100
Weish21	9073.52	98	9074	100	9074	100
Weish22	8938.18	51	8938.9	55	8938.36	52
Weish23	8343.67	89	8343.91	97	8343.94	98
Weish24	10220	100	10219.8	99	10220	100
Weish25	9934.66	73	9935.32	77	9935.16	76
Weish26	9574.93	73	9576.02	76	9575.16	73
Weish27	9819	100	9819	100	9819	100
Weish28	9492	100	9492	100	9492	100
Weish29	9410	100	9410	100	9410	100
Weish30	11190.2	81	11190.1	77	11190.3	82
PB1	3083.39	72	3087.6	84	3085.7	75
PB2	3178.33	69	3181.5	75	3181.09	77
PB4	95168	100	95168	100	95168	100
PB5	2133.53	70	2136.28	84	2138.32	96
PB6	775.58	97	776	100	776	100
PB7	1033.67	85	1034.17	91	1033.65	85
HP1	3410.74	65	3414.16	76	3414	76
HP2	3178.79	75	3183.48	89	3184.56	92
ผลรวม		4151		4289		4320

จากกราฟ ง.22 และ ง.23 ช่วงแรกของการค้นหาเส้นกราฟจะเพิ่มแล้วคงที่ เมื่อเส้นกราฟคงที่เป็นระยะเวลาจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีรันหรือรีโพซิชั่น หลังจากดำเนินการรีรันหรือรีโพซิชั่นเส้นกราฟจะมีค่าลดลงเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่ลงกว่าเดิม หลังจากนั้นจะดำเนินการค้นหาด้วย BPSO จึงทำให้เส้นกราฟเพิ่มขึ้น สำหรับกรณีปัญหาถุงเป้หลายมิตินั้น อนุภาคจะลู่กลับเข้ามาหา GBEST เดิม โดยที่ไม่พบตำแหน่งที่ดีกว่า GBEST เดิมในกรณี que เริ่มตันรีเซตที่จุด B สังเกตได้จากกราฟที่จะลู่ออกกลับมาในระดับเท่าเดิม เมื่อสังเกตระดับเส้นกราฟที่ลดลงหลังจากถูกรีเซต จะพบว่า RN มีระดับที่ลดลงน้อยกว่า RERUN เนื่องจาก RN ได้เปลี่ยนแปลงตำแหน่งของอนุภาคเพียงบางส่วน จึงเหลือส่วนที่เป็นผลจากการค้นหาเดิมอยู่ทำให้ตำแหน่งของอนุภาคที่ดีที่สุดแย่ลงกว่าเดิมไม่มากนัก เส้นกราฟจึงลดลงน้อยกว่าตอนเริ่มต้นค้นหา ในขณะที่ RERUN ทำการรีเซตอนุภาคทั้งหมดทำให้ตำแหน่งของอนุภาคทั้งหมดถูกกำหนดใหม่หมด เส้นกราฟจึงลดลงเทียบเท่ากับตอนเริ่มต้นค้นหา ทำให้ขอบเขตการค้นหาของ RERUN มากกว่า RN ซึ่งทำให้ต้องเสียเวลาในการลู่เข้าหากันของ RERUN มากกว่า RN จากตารางผลการทดลอง ง.8 จำนวนรอบของการรีเซตของ RN มีจำนวนมากกว่า RERUN หมายความว่ารีโพซิชั่นมีโอกาสในการลองค้นหา มากกว่ารีรัน จากตารางผลการทดลอง ง.9 RN ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า BPSO และ RERUN โดยสรุปแล้วปัญหาถุงเป้แบบหลายมิตินั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐานคือจำนวนรอบของการรีเซตของ RN มีจำนวนมากกว่า RERUN และ RN ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า BPSO และ RERUN

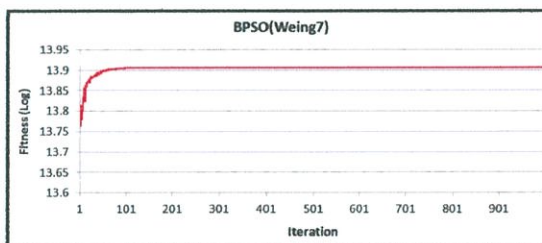
ง.6 การทดลองผลกระทบจากการรีเซต PBEST และ GBEST ของรีโพซิชั่น

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากการรีเซต GBEST และ PBEST ของรีโพซิชั่นกับปัญหาถุงเป้หลายมิติ จากกราฟ ง.24 ถึง ง.27 คือการทดลองรีโพซิชั่น โดยใช้พารามิเตอร์ตามตาราง 5.2 แต่จำนวนรอบการค้นหาคำตอบ 1000 รอบ และจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้ง เพราะแต่ละรอบของการทดลองการรีเซตเกิดขึ้นไม่พร้อมกัน ถ้าเป็นค่าเฉลี่ยจะเกิดการรวมค่าแล้วหักล้างกันทำให้ไม่พบผลของการรีเซต โดยกราฟทั้งสี่แสดงค่า LBP ของการค้นหาด้วย BPSO, RG, RP, RPG กับปัญหา WEING7 สำหรับตาราง ง.10 คือการทดลอง BPSO, RG, RP, RN, RPG กับปัญหาถุงเป้หลายมิติ โดยใช้พารามิเตอร์ตามตาราง 5.2 โดยตาราง ง.9 แสดงค่า MBF และ SR

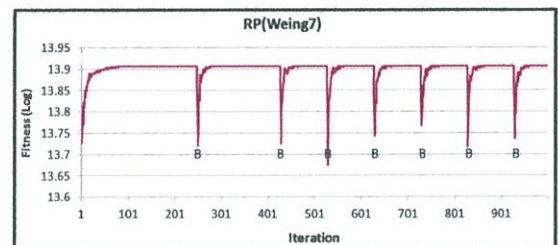
จากกราฟ ง.24 ช่วงแรกเส้นกราฟจะมีค่าเพิ่มขึ้นเนื่องจากผลจากการค้นหาด้วย BPSO และเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้นเส้นกราฟจะคงที่ตลอดจนสิ้นสุดการค้นหา จากกราฟ ง.23, ง.25, ง.26 ช่วงแรกของการค้นหาเส้นกราฟจะเพิ่มแล้วคงที่ เมื่อเส้นกราฟคงที่เป็นระยะเวลาจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีโพซิชั่นและหลังจากดำเนินการรีโพซิชั่น เส้นกราฟจะมีค่าลดลงเนื่องจากอนุภาคกระจายตัวออกทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่กว่าเดิม หลังจากนั้นจะดำเนินการค้นหาด้วย BPSO จึงทำให้เส้นกราฟเพิ่มขึ้น สำหรับกรณีปัญหาถุงเป้หลายมิตินั้น อนุภาคจะลู่กลับเข้ามาหา GBEST เดิม โดยที่ไม่พบตำแหน่งที่ดีกว่า GBEST เดิมในกรณี que เริ่มตันรีเซตที่จุด B สังเกตได้จากกราฟที่จะลู่ออกกลับมาในระดับเท่าเดิม

สำหรับจากกราฟ ง.27 ช่วงแรกของการค้นหาด้วย BPSO เส้นกราฟจะเพิ่มขึ้นและเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น เส้นกราฟจะคงที่เป็นระยะเวลาจนจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีโพรซิชั่นหลังจากดำเนินการรีโพรซิชั่นเส้นกราฟจะมีค่าลดลงเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่กว่าเดิม ซึ่งในบางครั้งอนุภาคก็ลู่เข้าหากันแล้วค้นหาในบริเวณใหม่ ซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นรีเซ็ตที่จุด C เส้นกราฟหลังจากการรีเซ็ตมีค่าเพิ่มขึ้นแล้วคงที่ในระดับที่ต่างจากเส้นกราฟก่อนการรีเซ็ต แต่ในบางครั้งอนุภาคก็ลู่เข้าค้นหาในบริเวณเดิม ซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นรีเซ็ตที่จุด D เส้นกราฟหลังจากการรีเซ็ตมีค่าเพิ่มขึ้นแล้วคงที่ในระดับที่เท่าเดิม จากพฤติกรรมการลู่เข้าหากันของ RPG กับการแก้ปัญหาหุงเป้แบบหลายมิติที่ได้อธิบายไปนี้ มีลักษณะเหมือนกับพฤติกรรมการลู่เข้าหากันของ RPG กับฟังก์ชันมาตรฐาน

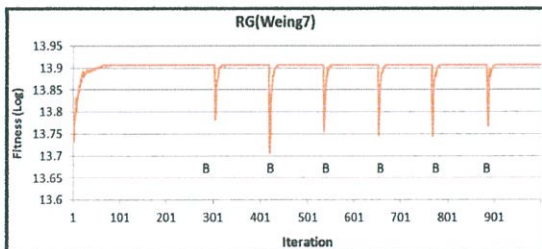
จากตารางผลการทดลอง ง.10 สำหรับ RPG ได้ผลลัพธ์โดยรวมดีที่สุดเมื่อเปรียบกับ BPSO, RP, RG, RN เพราะการค้นหาด้วย RPG นั้นอนุภาคสามารถมีโอกาสลู่เข้าไปค้นหาคำตอบในพื้นที่ใหม่ ซึ่งแตกต่างจาก RN, RP, RG ที่ลู่เข้าหาพื้นที่เดิม ทำให้ RPG มีขอบเขตการค้นหาที่กว้างกว่า RN, RP, RG โดยสรุปแล้วปัญหาหุงเป้แบบหลายมิตินั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ RPG ได้ผลลัพธ์โดยรวมดีที่สุดเมื่อเปรียบกับ BPSO, RP, RG, RN



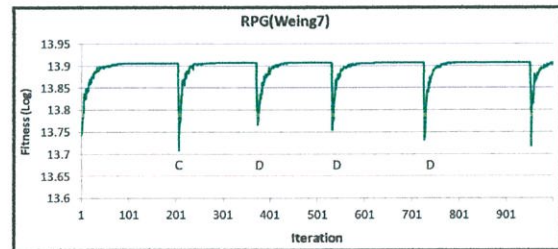
รูปที่ ง.24 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย BPSO



รูปที่ ง.25 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย RP



รูปที่ ง.26 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย RG



รูปที่ ง.27 LBP ปัญหา WEING7 ที่ค้นหาคำตอบด้วย RPG

ตารางที่ ง.10 ผลการค้นหาคำตอบของปัญหาถุงเบ้หลายมิติด้วย BPSO, RG, RP, RN, RPG

Algorithm	BPSO		RN		RP		RG		RPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
Sento1	7767.82	75	7769.62	83	7770.74	91	7768.89	79	7770.46	89
Sento2	8716.29	37	8718.53	59	8718.93	70	8717.91	59	8721.7	94
Weing1	141278	100	141278	100	141278	100	141278	100	141278	100
Weing2	130883	100	130883	100	130883	100	130883	100	130883	100
Weing3	95643.7	91	95677	100	95677	100	95677	100	95677	100
Weing4	119337	100	119337	100	119337	100	119337	100	119337	100
Weing5	98796	100	98796	100	98796	100	98796	100	98796	100
Weing6	130611	97	130623	100	130623	100	130623	100	130623	100
Weing7	1095330	9	1095350	11	1095380	14	1095350	8	1095440	95
Weing8	621019	13	621242	16	621289	12	621045	8	623284	68
Weish01	4554	100	4554	100	4554	100	4554	100	4554	100
Weish02	4535.3	86	4535.1	82	4535.5	90	4535.3	86	4535.25	85
Weish03	4115	100	4115	100	4115	100	4115	100	4115	100
Weish04	4561	100	4561	100	4561	100	4561	100	4561	100
Weish05	4514	100	4514	100	4514	100	4514	100	4514	100
Weish06	5553.24	78	5555.63	91	5557	100	5556.02	94	5557	100
Weish07	5567	100	5567	100	5567	100	5567	100	5567	100
Weish08	5604.94	97	5605	100	5605	100	5605	100	5605	100
Weish09	5246	100	5246	100	5246	100	5246	100	5246	100
Weish10	6339	100	6339	100	6339	100	6339	100	6339	100
Weish11	5643	100	5643	100	5643	100	5643	100	5643	100
Weish12	6339	100	6339	100	6339	100	6339	100	6339	100
Weish13	6159	100	6159	100	6159	100	6159	100	6159	100
Weish14	6954	100	6954	100	6954	100	6954	100	6954	100
Weish15	7486	100	7486	100	7486	100	7486	100	7486	100
Weish16	7288.88	94	7289	100	7289	100	7289	100	7289	100
Weish17	8633	100	8633	100	8633	100	8633	100	8633	100
Weish18	9578.35	87	9580	100	9580	100	9580	100	9580	100
Weish19	7696.83	91	7697.48	96	7698	100	7697.74	98	7698	100
Weish20	9449.6	98	9450	100	9450	100	9450	100	9450	100
Weish21	9073.52	98	9074	100	9074	100	9074	100	9074	100
Weish22	8938.18	51	8938.36	52	8940.52	64	8936.02	39	8946.64	98
Weish23	8343.67	89	8343.94	98	8344	100	8343.94	98	8344	100
Weish24	10220	100	10220	100	10220	100	10220	100	10220	100
Weish25	9934.66	73	9935.16	76	9938.36	96	9936.12	82	9939	100
Weish26	9574.93	73	9575.16	73	9577.84	82	9578.16	83	9584	100
Weish27	9819	100	9819	100	9819	100	9819	100	9819	100
Weish28	9492	100	9492	100	9492	100	9492	100	9492	100
Weish29	9410	100	9410	100	9410	100	9410	100	9410	100
Weish30	11190.2	81	11190.3	82	11190.5	87	11190.2	79	11191	100
PB1	3083.39	72	3085.7	75	3087.48	82	3086.64	76	3088.46	89
PB2	3178.33	69	3181.09	77	3183.48	86	3182.89	87	3185.46	97
PB4	95168	100	95168	100	95168	100	95168	100	95168	100
PB5	2133.53	70	2138.32	96	2139	100	2138.66	98	2139	100
PB6	775.58	97	776	100	776	100	776	100	776	100
PB7	1033.67	85	1033.65	85	1034.64	96	1033.92	88	1034.91	99
HP1	3410.74	65	3414	76	3413.78	71	3413.5	69	3416.32	88
HP2	3178.79	75	3184.56	92	3184.02	89	3184.74	93	3185.64	98
ผลรวม		4151		4320		4430		4324		4700

ภาคผนวก จ

การทดลองการประยุกต์การกลายพันธุ์และรีโพอิชันกับ TSP

ในภาคผนวกนี้จะดำเนินการทดลองเพื่ออธิบายแนวคิดที่มาจากขั้นตอนวิธีที่นำเสนอ (MRPSO) เมื่อนำไปประยุกต์ใช้กับปัญหาการเดินทางของพนักงานขาย ซึ่งเป็นปัญหาในรูปแบบการสลับลำดับกันในพีชคณิตจึงมีคุณสมบัติแตกต่างจากปัญหาการหาจุดที่ดีที่สุดเชิงต่อเนื่องที่ได้ดำเนินการทดลองในบทที่ 5

โดยการทดลองในภาคผนวกนี้จะพิสูจน์ว่าผลลัพธ์จากการค้นหาคำตอบด้วยรูปแบบการสลับลำดับกันในพีชคณิตจะให้ผลลัพธ์เหมือนกับฟังก์ชันมาตรฐานในบทที่ 5 สำหรับรายละเอียดขั้นตอนวิธีที่นำมาทดลองถูกกล่าวไว้ในบทที่ 5 และรายละเอียดชุดการทดลองถูกกล่าวไว้ในบทที่ 5 ส่วนการทดลองสามารถแบ่งออกเป็น 6 ส่วน โดยพิจารณาจากจุดประสงค์การทดลองมีรายละเอียดดังนี้

- การทดลองปัญหาการติดในจุดสุดสัมพัทธ์ของ PPSO โดยทดลองกับปัญหาการเดินทางของพนักงานขาย
- การทดลองการประยุกต์ใช้การกลายพันธุ์กับ PPSO โดยทดลองกับปัญหาการเดินทางของพนักงานขาย
- การทดลองปัญหาจากการกลายพันธุ์แบบ NCM โดยทดลองกับปัญหาการเดินทางของพนักงานขาย
- การทดลองปัญหาจากการกลายพันธุ์และแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM โดยทดลองกับปัญหาการเดินทางของพนักงานขาย
- การทดลองการประยุกต์ใช้รีรันและรีโพอิชันโดยทดลองกับปัญหาการเดินทางของพนักงานขาย
- การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพอิชันโดยทดลองกับปัญหาการเดินทางของพนักงานขาย

ตารางที่ จ.1 สรุปสมบัติของปัญหาการเดินทางของพนักงานขายที่ใช้ในการทดลอง

ชื่อปัญหา	ค่าที่ดีที่สุด	จำนวน โหนด	จำนวนอนุภาค (กรณีไม่ ดำเนินการ CM)	จำนวนอนุภาค (กรณีดำเนินการ CM)	จำนวนครั้งใน การทดลอง	รอบของการกลาย พันธุ์ของขั้นตอนวิธี CM
BURMA14	3323	14	420	20	100	20
ULYSSES16	6859	16	420	20	100	20
GR17	2085	17	420	20	100	20
GR21	2707	21	420	20	100	20
ULYSSES22	7013	22	420	20	100	20
FRI26	937	26	420	20	100	20
BAYG29	1610	29	420	20	100	20
BAYS29	2020	29	420	20	100	20
DANTZIG42	699	42	40200	200	10	200
SWISS42	1273	42	40200	200	10	200
GR48	5046	48	40200	200	10	200
EIL51	426	51	40200	200	10	200
BERLIN52	7542	52	40200	200	10	200
BRAZIL58	25395	58	40200	200	10	200

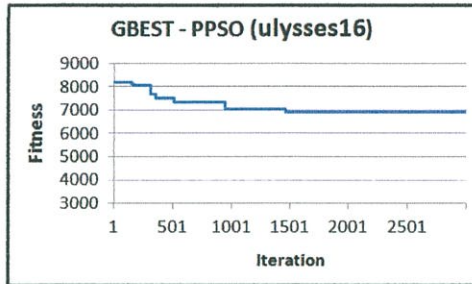
จ.1 การทดลองปัญหาการติดในจุดสุดสัมพัทธ์

ในหัวข้อนี้วิเคราะห์ผลกระทบจากปัญหาการเดินทางของพนักงานขายที่เกิดกับ PPSO โดยตาราง จ.2 ได้ใช้พารามิเตอร์ตามตาราง 5.3 แต่สำหรับตารางนี้ได้ใช้จำนวนรอบการค้นหา 5000 รอบ และ 50000 รอบ โดยตาราง จ.2 แสดงค่า MBF และ SR

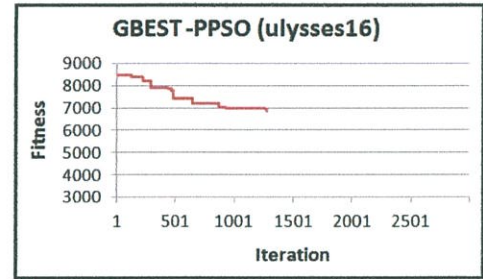
ตารางที่ จ.2 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO

ขั้นตอนวิธี	PPSO (5000)		PPSO (50000)	
	MBF	SR	MBF	SR
BURMA14	3326.77	71	3327.68	64
ULYSSES16	6880.84	37	6878.27	35
GR17	2096.54	64	2093.47	73
GR21	2813.44	39	2806	46
ULYSSES22	7128.44	8	7115.74	14
FRI26	971.49	1	975.55	1
BAYG29	1770.9	0	1764.02	0
BAYS29	2090.16	1	2096.35	1
ผลรวม		221		234

จากตาราง ง.2 โดยภาพรวมแล้ว PPSO ที่มีจำนวนรอบการค้นหาคำตอบมาก (50000 รอบ) จะได้รับผลลัพธ์การค้นหาคำตอบดีกว่า PPSO ที่มีจำนวนรอบการค้นหาคำตอบน้อย (5000 รอบ) แต่ในบางปัญหาจำนวนรอบการค้นหาคำตอบที่น้อยกว่าก็ได้ผลลัพธ์ที่ดีกว่าจำนวนรอบการค้นหาคำตอบที่มาก เช่น ULYSSES16 และ BURMA14 ที่เป็นเช่นนี้เพราะ PPSO เป็นปัญหาหมัดติโมดอล ซึ่งในบางครั้งการค้นหาคำตอบประสบปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น การเพิ่มจำนวนรอบการค้นหาคำตอบให้มากขึ้นก็ไม่สามารถแก้ปัญหาการติดในจุดสุดสัมพัทธ์ได้



รูปที่ จ.1 ค่าความเหมาะสมของ PPSO ที่เกิดปัญหาการติดในจุดสุดสัมพัทธ์



รูปที่ จ.2 ค่าความเหมาะสมของ PPSO ที่ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์

เพื่อแสดงให้เห็นผลกระทบจากปัญหาการติดในจุดสุดสัมพัทธ์ จึงได้ทำการทดลองโดยการค้นหาคำตอบด้วย PPSO กับปัญหา ULYSSES16 โดยใช้พารามิเตอร์ตามตาราง 5.3 แต่กราฟ จ.1 และ จ.2 ได้กำหนดจำนวนรอบการค้นหาคำตอบ 3000 รอบ และกราฟแสดงค่าความเหมาะสมของ GBEST โดยการค้นหานั้นจะดำเนินการเพียงครั้งเดียว

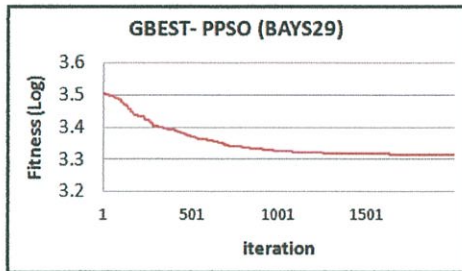
จากผลการทดลองสรุปได้ว่า การค้นหาคำตอบกับปัญหาการเดินทางของพนักงานขายด้วย PPSO แม้จะเป็นปัญหาเดียวกัน แต่ในบางครั้งการค้นหาก็ประสบปัญหาการติดในจุดสุดสัมพัทธ์ ดังกราฟ จ.1 โดยหลังจากการค้นหาผ่านไประยะเวลาหนึ่ง คำตอบที่ได้จะไม่ดีขึ้นเลยและไม่ว่าเวลาจะผ่านไปนานเท่าใด คำตอบที่ได้ก็ไม่มีแนวโน้มว่าจะดีขึ้น แต่ในบางครั้งก็ไม่เกิดปัญหาการติดในจุดสุดสัมพัทธ์ซึ่งจะทำให้พบคำตอบที่ดีที่สุดดังกราฟ จ.2 โดยสรุปแล้วผลกระทบจากปัญหาการติดในจุดสุดสัมพัทธ์ของปัญหาการเดินทางของพนักงานขายให้ผลลัพธ์เหมือนกับฟังก์ชันมาตรฐาน

จ.2 การทดลองการประยุกต์ใช้การกลายพันธุ์

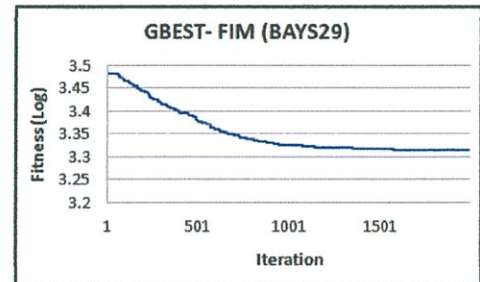
ในหัวข้อนี้พิจารณาถึงความสามารถในการค้นหาคำตอบด้วย PPSO, MXUX ซึ่งเป็นตัวแทนของการกลายพันธุ์แบบ CM และ FIM ซึ่งเป็นตัวแทนของการกลายพันธุ์แบบ NCM เพื่อวิเคราะห์ผลกระทบจากการกลายพันธุ์กับ PPSO ที่ดำเนินการแก้ปัญหาการเดินทางของพนักงานขาย

จ.2.1 การพิจารณาการกลายพันธุ์แบบ NCM

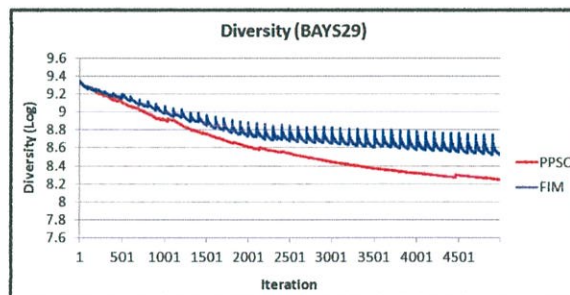
หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย PPSO และ FIM กับปัญหา BAYS29 สำหรับกราฟ จ.3 และ จ.4 ใช้พารามิเตอร์ตามตาราง 5.3 แต่ใช้จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟแสดงค่า LMFG ส่วนกราฟ จ.5 ใช้พารามิเตอร์ตามตาราง 5.3 แต่ใช้จำนวนรอบการค้นหาคำตอบ 5000 รอบ โดยกราฟแสดงค่า LD (t) โดย LD (t) คือค่าลอการิทึมของค่าเฉลี่ยความหลากหลายของประชากรในแต่ละรอบการค้นหาคำตอบ สำหรับตารางการทดลอง จ.3 ใช้พารามิเตอร์ตามตาราง 5.3 โดยตาราง จ.3 แสดงค่า MBF และ SR



รูปที่ จ.3 LMFG ของปัญหา BAYS29 ที่ค้นหาคำตอบด้วย PPSO

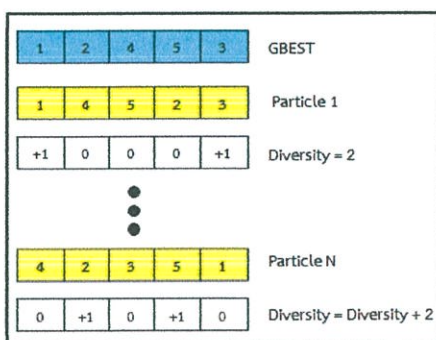


รูปที่ จ.4 LMFG ของปัญหา BAYS29 ที่ค้นหาคำตอบด้วย FIM



รูปที่ จ.5 LD(t) ปัญหา BAYS29 ที่ค้นหาคำตอบด้วย PPSO และ FIM

เนื่องจากปัญหาการเดินทางของพนักงานขายไม่มีการคำนวณความหลากหลายของประชากร ดังนั้นงานวิจัยฉบับนี้ได้เสนอให้มีการคำนวณความหลากหลายโดยใช้วิธีการเปรียบเทียบความแตกต่างระหว่างตำแหน่งที่ดีที่สุดที่อนุภาคเคยค้นพบหรือ GBEST กับอนุภาคทั้งหมด ถ้าหากมิติบนอนุภาคมีลำดับของโนดที่แตกต่างจาก GBEST ค่าความหลากหลายก็จะเพิ่มขึ้นดังรูป จ.6 แต่ถ้าหากมิติที่อยู่บนอนุภาคมีมิติที่เหมือนกับ GBEST ค่าความหลากหลายจะไม่เพิ่มขึ้นดังรูป จ.6 เมื่อดำเนินการเช่นนี้จนครบทุกอนุภาค ผลลัพธ์ที่ได้ก็จะเป็นความหลากหลายของประชากรในรอบการค้นหาคำตอบ



รูปที่ จ.6 ตัวอย่างการคำนวณความหลากหลายของประชากรของ PPSO

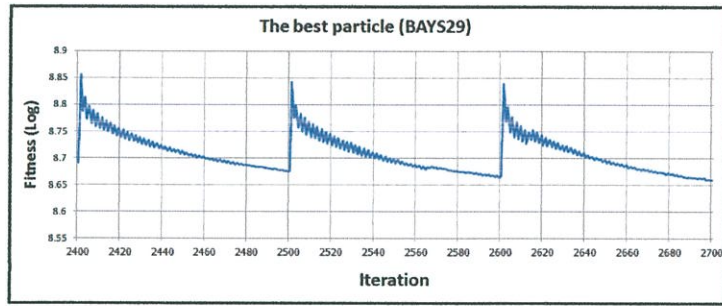
ตารางที่ จ.3 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO และ FIM

ขั้นตอนวิธี	PPSO		FIM	
	MBF	SR	MBF	SR
burma14	3327.68	64	3323	100
ulysses16	6878.27	35	6860.9	85
gr17	2093.47	73	2085.6	90
gr21	2806	46	2736.92	76
ulysses22	7115.74	14	7068	40
fri26	975.55	1	963.05	8
bayg29	1764.02	0	1700.49	0
bays29	2096.35	1	2071	3
ผลรวม		234		402

จากกราฟ จ.3 และ จ.4 กราฟทั้งสองเกิดปัญหาการติดในจุดสุดสัมพัทธ์ โดยสังเกตจากเส้นกราฟมีค่าคงที่ สำหรับกราฟ จ.5 แสดงให้เห็นว่า PPSO และ FIM มีความหลากหลายของประชากรลดลง โดย PPSO มีความหลากหลายของประชากรน้อยกว่า FIM ส่งผลให้ FIM มีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ได้มากกว่า PPSO จากผลการทดลองตาราง จ.3 พบว่า FIM ได้ผลลัพธ์โดยรวมดีกว่า PPSO เพราะ FIM สามารถเพิ่มความหลากหลายให้กับประชากรจึงสามารถลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า PPSO โดยสรุปแล้วปัญหาการเดินทางของพนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ FIM มีความหลากหลายมากกว่าและได้รับคำตอบจากการค้นหาที่ดีกว่า PPSO

จ.2.2 การพิจารณาการเพิ่มประสิทธิภาพการค้นหาของ PSO ด้วย NCM

หัวข้อนี้ได้ทดลองการค้นหาคำตอบด้วย FIM กับปัญหา BAYS29 โดยกราฟ จ.7 ได้ใช้พารามิเตอร์ตามตาราง 5.3 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหาคำตอบ 3000 รอบ และจำนวนครั้งในการทดลอง 1 ครั้งเท่านั้น โดยกราฟแสดงค่าของ LBP ของการค้นหาด้วย FIM กับปัญหา BAYS29 ในรอบการค้นหาช่วง 2400 ถึง 2700 รอบ



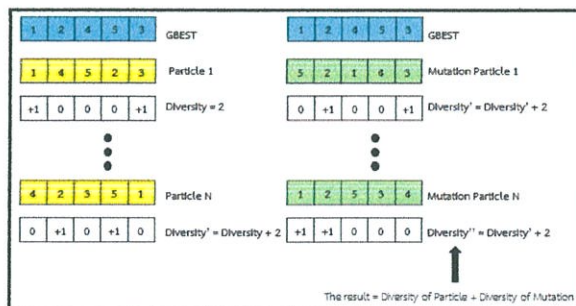
รูปที่ จ.7 FIM ปรับปรุง PPSO ในช่วงรอบการค้นหา 2400 ถึง 2700

จากรูป จ.7 การกลายพันธุ์ถูกกำหนดให้ดำเนินการทุก 100 รอบของการค้นหาโดยการกลายพันธุ์ทำให้เกิดการกระจายตัวของอนุภาคออกจากตำแหน่งของ GBEST ไปยังตำแหน่งที่แย่กว่าเดิมสังเกตได้จากเส้นกราฟเมื่อผ่านการกลายพันธุ์เส้นกราฟจะมีค่าเพิ่มขึ้น หลังจากนั้นอนุภาคจะทำการลู่เข้าหากันใหม่ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้คำตอบนั้นดีขึ้นสังเกตได้จากเส้นกราฟมีค่าลดลงมากกว่าก่อนการกลายพันธุ์ จากผลการทดลองสรุปว่าการกลายพันธุ์ทำให้อนุภาคกระจายตัวออกจากกันแล้วการปรับปรุงคำตอบเกิดจากการที่อนุภาคลู่เข้าหากัน ซึ่งผลกระทบจากการปรับปรุงคำตอบ NCM ที่ใช้กับปัญหาการเดินทางของพนักงานขายเหมือนกับผลกระทบจากการปรับปรุงคำตอบของ NCM ที่ใช้กับฟังก์ชันมาตรฐาน

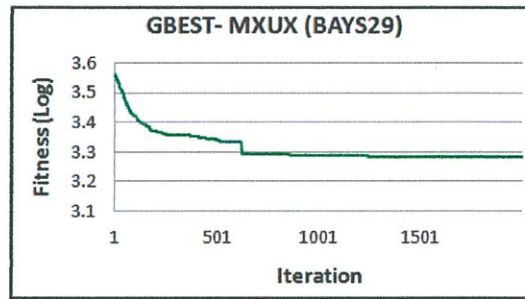
จ.2.3 การพิจารณาการกลายพันธุ์แบบ CM

หัวข้อนี้ได้ทดลองค้นหาคำตอบด้วย PPSO และ MXUX กับปัญหา BAY29 โดยกราฟ จ.9 ใช้พารามิเตอร์ตามตาราง 5.3 แต่จำนวนรอบการค้นหาคำตอบ 2000 รอบ โดยกราฟนี้ได้แสดงค่า LMFG ของ MXUX สำหรับกราฟ จ.10 ใช้พารามิเตอร์ตามตาราง 5.3 แต่จำนวนรอบการค้นหาคำตอบ 5000 รอบ ได้แสดงค่า LD (t) ของการค้นหาด้วย PPSO และ FIM สำหรับตารางการทดลอง จ.4 ใช้พารามิเตอร์ตามตาราง 5.3 โดยตาราง จ.4 แสดงค่า MBF และ SR

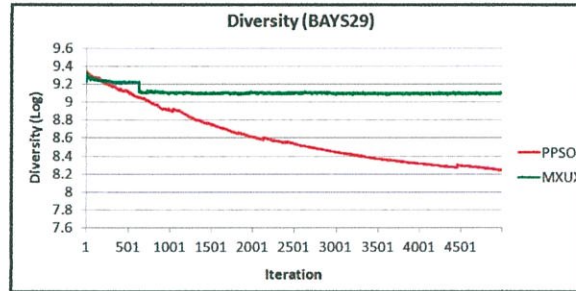
จากหัวข้อ จ.2.1 ได้อธิบายเกี่ยวกับการคำนวณความหลากหลายของประชากรในกรณี PPSO และ FIM ซึ่ง MXUX ก็มีขั้นตอนการคำนวณความหลากหลายของประชากรเหมือนกัน PPSO และ FIM เพียงแต่กรณี MXUX ต้องมีขั้นตอนการคำนวณความหลากหลายของประชากรเพิ่มเติมคือต้องเปรียบเทียบความแตกต่างระหว่างตำแหน่งที่ดีที่สุดที่อนุภาคเคยค้นพบหรือ GBEST กับตำแหน่งที่เกิดจากการดำเนินการกลายพันธุ์ดังรูป จ.8 โดยการคำนวณความหลากหลายของประชากรใช้วิธีการคำนวณเหมือนกับกรณี PPSO และ FIM



รูปที่ จ.8 ตัวอย่างการคำนวณความหลากหลายของประชากรของ MXUX



รูปที่ จ.9 LMFG ของปัญหา BAY29 ด้วย MXUX



รูปที่ จ.10 LD(t) ของปัญหา BAY29 ด้วย PPSO และ MXUX

ตารางที่ จ.4 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO และ MXUX

ขั้นตอนวิธี	PPSO		MXUX	
	MBF	SR	MBF	SR
burma14	3327.68	64	3323	100
ulysses16	6878.27	35	6859	100
gr17	2093.47	73	2085	100
gr21	2806	46	2707	100
ulysses22	7115.74	14	7032.22	75
fri26	975.55	1	957.73	30
bayg29	1764.02	0	1677.87	3
bays29	2096.35	1	2092.82	4
ผลรวม		234		512

จากกราฟ จ.9 เกิดปัญหาการติดในจุดสุดสัมพัทธ์โดยสังเกตจากเส้นกราฟมีค่าคงที่ จากกราฟ จ.10 จะเห็นว่าเส้นกราฟของ PPSO มีความหลากหลายประชากรน้อยกว่า MXUX ส่งผลให้ MXUX มีขอบเขตการค้นหาที่กว้างกว่า PPSO จึงมีโอกาสกระโดดออกจากจุดสุดสัมพัทธ์ได้มากกว่า

จากตาราง จ.4 พบว่า MXUX ได้ผลลัพธ์โดยรวมดีกว่า PPSO เพราะ MXUX สามารถเพิ่มความหลากหลายให้กับประชากรจึงสามารถลดปัญหาการติดในจุดสุดสัมพัทธ์ได้ดีกว่า PPSO ดังนั้นการกลายพันธุ์แบบ CM สามารถช่วยปรับปรุงคำตอบและเพิ่มประสิทธิภาพการค้นหาของ PPSO โดยสรุปแล้วปัญหาการเดินทางของพนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐานคือ MXUX มีความหลากหลายมากกว่าและได้รับคำตอบจากการค้นหาที่ดีกว่า PPSO

จ.3 การทดลองปัญหาจากการกลายพันธุ์แบบ NCM

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากปัญหาที่เกิดจากการกลายพันธุ์แบบ NCM กับปัญหาการเดินทางของพนักงานขาย

จ.3.1 การพิจารณาปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบ

จากตารางผลการทดลอง จ.5 ได้ค้นหาคำตอบด้วย PPSO และ FIM กับปัญหาการเดินทางของพนักงานขาย โดยใช้พารามิเตอร์ตามตาราง 5.3 แต่สำหรับการทดลองนี้รอบของการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 1 รอบ ความน่าจะเป็นในการกลายพันธุ์ของ FIM ได้กำหนดค่าเท่ากับ 0.40 โดยตาราง จ.5 แสดงค่า MBF และ SR

ตารางที่ จ.5 ผลการทดลองปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบสำหรับ TSP

ขั้นตอนวิธี	PPSO		FIM	
	MBF	SR	MBF	SR
burma14	3327.68	64	3323	100
ulysses16	6878.27	35	6913.18	6
gr17	2093.47	73	2153.04	1
gr21	2806	46	3380.88	0
ulysses22	7115.74	14	8138.81	0
fri26	975.55	1	1327.55	0
bayg29	1764.02	0	2551.7	0
bays29	2096.35	1	3079.12	0
ผลรวม		234		107

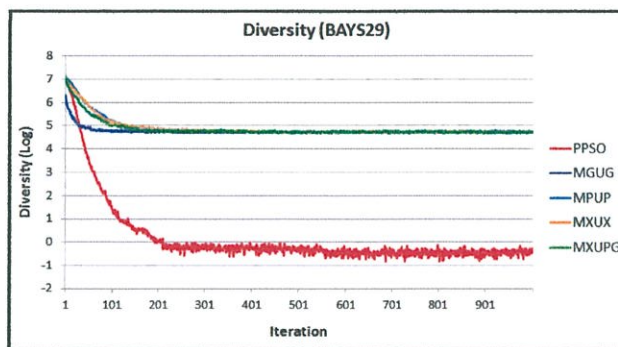
จากผลการทดลองตารางนี้แสดงให้เห็นว่า ผลลัพธ์ที่ได้มีลักษณะเหมือนกรณีของฟังก์ชันมาตรฐาน คือการกำหนดให้ดำเนินการกลายพันธุ์มากเกินไป (1 รอบ และ 30 เปอร์เซ็นต์) ผลลัพธ์จากการค้นหาของ FIM โดยรวมแล้วแยกว่า PPSO

จ.4 การทดลองปัญหาและแนวทางแก้ปัญหาการกลายพันธุ์แบบ CM

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากปัญหาที่เกิดจากการกลายพันธุ์แบบ CM และแนวทางการแก้ปัญหาที่เกิดจากการกลายพันธุ์แบบ CM กับปัญหาการเดินทางของพนักงานขาย

จ.4.1 การพิจารณาความหลากหลายของประชากรของการกลายพันธุ์แบบ CM

สำหรับกราฟ จ.11 คือการทดลองค้นหาคำตอบด้วย PPSO, MXUX, MPUP, MGUG และ MXUPG กับปัญหา BAYS29 โดยใช้พารามิเตอร์ตามตาราง 5.3 แต่จำนวนรอบการค้นหาคำตอบ 1000 รอบ โดยกราฟแสดงค่า LD(t) ของการค้นหา

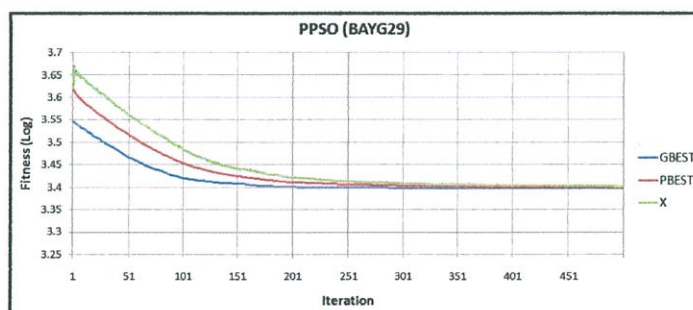


รูปที่ จ.11 LD(t) ของ PPSO, MGUG, MPUP, MXUX และ MXUPG ของปัญหา BAYS29

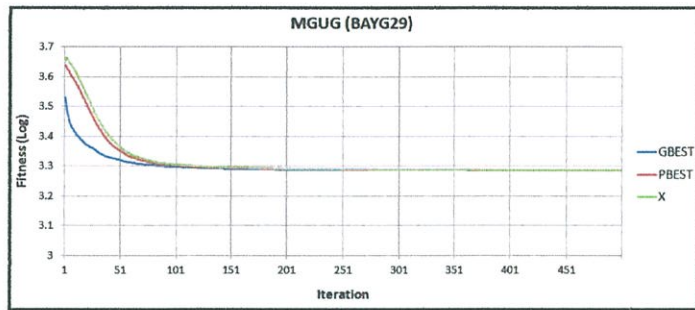
จากกราฟ จ.11 ความหลากหลายของประชากรในช่วงเริ่มต้นของการค้นหาของ PPSO, MGUG, MPUP, MXUX และ MXUPG จะมีค่ามาก เมื่อเวลาผ่านไปความหลากหลายของประชากรจะมีค่าลดลง เพราะอนุภาคพยายามลู่เข้าหากันตามหลักการทำงานของ PPSO และเมื่ออนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ความหลากหลายของประชากรจะอยู่ในระดับคงที่ ในช่วงที่ระดับคงที่เส้นกราฟของ MGUG, MPUP, MXUX และ MXUPG จะมีระดับใกล้เคียงกันและมีระดับที่มากกว่า PPSO หมายความว่าในช่วงที่อนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ความหลากหลายของประชากรของ MGUG, MPUP, MXUX และ MXUPG จะมีค่าใกล้เคียงกันและมีค่ามากกว่าความหลากหลายของประชากรของ PPSO โดยสรุปแล้วปัญหาการเดินทางของพนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MGUG, MPUP, MXUX และ MXUPG มีความหลากหลายมากกว่า PPSO

จ.4.2 การพิจารณาปัญหาการลู่เร็วเกินไปและแนวทางแก้ปัญหาลู่เร็วเกินไป

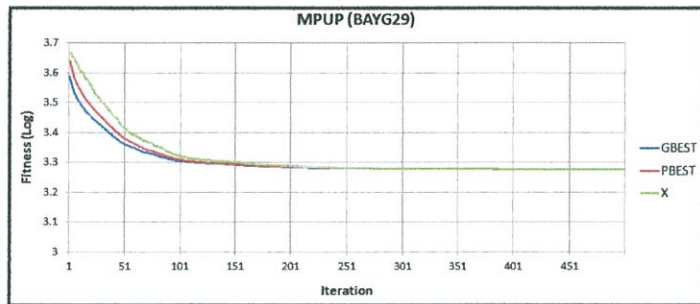
กราฟ จ.12 ถึง จ.16 ได้ทดลองค้นหาคำตอบด้วย BPSO, MXUX, MPUP, MGUG และ MXUPG กับปัญหา BAYG29 โดยใช้พารามิเตอร์ตามตาราง 5.3 แต่ได้กำหนดจำนวนรอบการค้นหาคำตอบ 500 รอบ โดยกราฟแสดงค่า LMFx, LMFP และ LMFG ส่วนตาราง จ.6 ได้กำหนดจำนวนอนุภาคให้เท่ากับ 5 อนุภาค ความน่าจะเป็นในการกลายพันธุ์ได้กำหนดค่าเท่ากับ 0.80 และรอบของการกลายพันธุ์เท่ากับ 84 ส่วนตาราง จ.7 โดยใช้พารามิเตอร์ตามตาราง 5.3 โดยตาราง จ.6 และ จ.7 แสดงค่า MBF และ SR



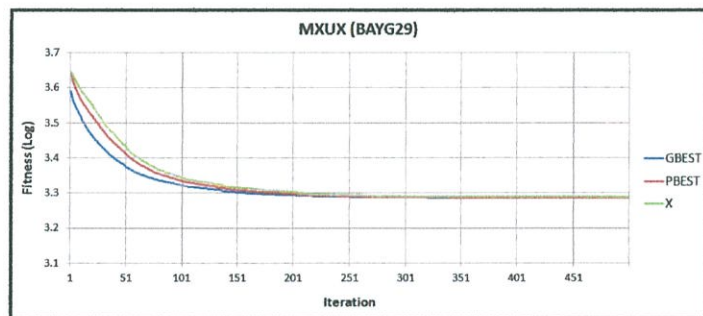
รูปที่ จ.12 LMFx, LMFP และ LMFG ของ PPSO ของปัญหา BAYG29



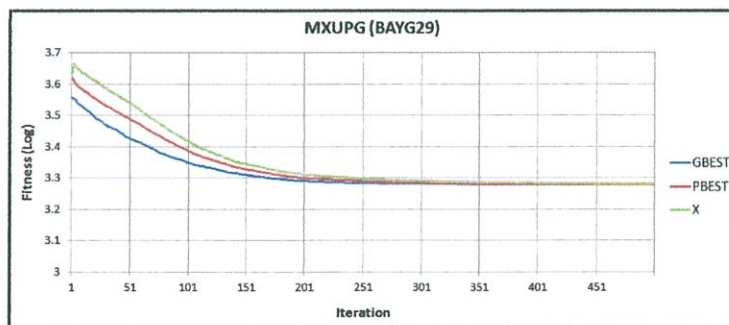
รูปที่ จ.13 LMFx, LMFP และ LMFG ของ MGUG ของปัญหา BAYG29



รูปที่ จ.14 LMFx, LMFP และ LMFG ของ MPUP ของปัญหา BAYG29



รูปที่ จ.15 LMFx, LMFP และ LMFG ของ MXUX ของปัญหา BAYG29



รูปที่ จ.16 LMFx, LMFP และ LMFG ของ MXUPG ของปัญหา BAYG29

จากกราฟ จ.12 ถึง จ.16 ในช่วงแรกของการค้นหาเส้นกราฟของ LMFx, LMFP, LMFG ของ PPSO, MGUG, MPUP, MXUX, MXUPG จะมีค่าลดลง เพราะอนุภาคพยายามลู่เข้าหากันตามหลักการทำงานของ PPSO และเมื่ออนุภาคนั้นติดอยู่ในจุดสุดสัมพัทธ์ เส้นกราฟทั้ง 3 จะเข้าสู่สถานะคงที่จนสิ้นสุดการค้นหา สำหรับเส้นกราฟของ LMFx ของ MGUG, MPUP, MXUX รอบของการค้นหาเข้าสู่สถานะคงที่ประมาณ 100, 200 และ 250 รอบตามลำดับ ในขณะที่เส้นกราฟของ LMFx ของ PPSO และ MXUPG รอบของการค้นหาเข้าสู่สถานะคงที่ประมาณ 320 รอบ และ 300 ตามลำดับ ซึ่ง MGUG, MPUP, MXUX เข้าสู่สภาพคงที่เร็วกว่า PPSO และ MXUPG หมายความว่า MGUG, MPUP, MXUX มีความเร็วในการลู่เข้าหากันเร็วกว่า PPSO และ MXUPG ซึ่งทำให้ MGUG, MPUP, MXUX เกิดปัญหาการลู่เร็วเกินไปได้ง่ายกว่า PPSO และ MXUPG โดยสรุปแล้วปัญหาการเดินทางของพนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MGUG, MPUP, MXUX มีความเร็วในการลู่เข้าหากันมากกว่า PPSO และ MXUPG

ตารางที่ จ.6 ผลการทดลองปัญหาการลู่เร็วเกินไปสำหรับ TSP

ขั้นตอนวิธี	PPSO		MXUX (มาก)	
	MBF	SR	MBF	SR
burma14	3327.68	64	3328.36	71
ulysses16	6878.27	35	6976.73	4
gr17	2093.47	73	2190.73	0
gr21	2806	46	3612.15	0
ulysses22	7115.74	14	8138.11	0
fri26	975.55	1	1400.53	0
bayg29	1764.02	0	2660.54	0
bays29	2096.35	1	3362.42	0
ผลรวม		234		75

ตารางที่ จ.7 ผลการค้นหาคำตอบของ TSP ด้วย PPSO, MGUG, MXUX, MPUP และ MXUPG

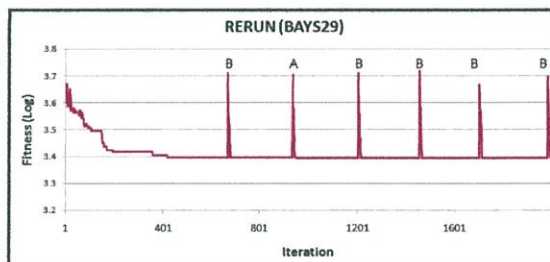
ขั้นตอนวิธี	PPSO		MGUG		MPUP		MXUX		MXUPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
burma14	3327.68	64	3323	100	3323	100	3323	100	3323	100
ulysses16	6878.27	35	6859	100	6859	100	6859	100	6859	100
gr17	2093.47	73	2085	100	2085	100	2085	100	2085	100
gr21	2806	46	2707	100	2707	100	2707	100	2707	100
ulysses22	7115.74	14	7034.34	72	7035.83	69	7032.22	75	7032.38	77
fri26	975.55	1	958.45	20	950.36	33	957.73	30	949.24	30
bayg29	1764.02	0	1672.28	3	1654.04	1	1677.87	3	1627.34	38
bays29	2096.35	1	2092.7	1	2048.21	7	2092.82	4	2032.56	23
ผลรวม		234		496		510		512		568

จากตารางผลการทดลอง จ.6 ผลลัพธ์การค้นหาโดยรวมของ MXUX แย่กว่า PPSO เนื่องจาก MXUX เกิดปัญหาการลู่เร็วเกินไปมากกว่า PPSO ปกติ จึงทำให้ผลลัพธ์การค้นหาคำตอบ แย่กว่า PPSO จากตารางผลการทดลอง จ.7 ผลลัพธ์การค้นหาโดยรวมของ MPUP, MXUX, MGUG ดีกว่า PPSO และโดยรวม MXUPG ได้ผลลัพธ์ที่ดีที่สุด โดยสาเหตุที่ MXUPG สามารถหาคำตอบได้ดี ที่สุดนั้นเป็นผลมาจากที่ MXUPG ไม่ทำให้ปัญหาการลู่เร็วเกินไปเกิดขึ้นมากกว่า PPSO ปกติ จึงทำให้ผลลัพธ์การค้นหาคำตอบดีกว่ากับ MXUX, MGUG, MPUP โดยสรุปแล้วปัญหาการเดินทางของ พนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ MXUPG ได้ผลลัพธ์ที่ดีที่สุดเมื่อเปรียบเทียบกับ MGUG, MPUP, MXUX และ PPSO และเมื่อกำหนดให้ดำเนินการกลายพันธุ์มากขึ้นไป MXUX จะได้ผลลัพธ์การค้นหาแย่กว่า PPSO

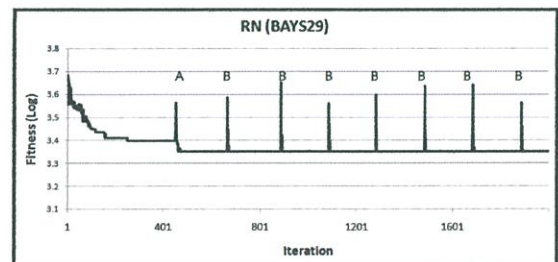
จ.5 การทดลองการประยุกต์ใช้รีรันและรีโพลิชัน

ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากรีรันและรีโพลิชันกับปัญหาการเดินทางของพนักงานขาย โดยการทดลองนี้ได้เลือกใช้ RN เป็นตัวแทนของรีโพลิชัน เพราะทั้ง RN และ RERUN ไม่มีขั้นตอนการรีเซ็ตทั้ง GBEST และ PBEST

จากกราฟ จ.17 และ จ.18 คือ การทดลองรีโพลิชันและรีรันโดยใช้พารามิเตอร์ตามตาราง 5.3 แต่จำนวนรอบการค้นหาคำตอบ 2000 รอบ และจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้ง เพราะแต่ละรอบของการทดลองการรีเซ็ตเกิดขึ้นไม่พร้อมกัน ถ้าเป็นค่าเฉลี่ยจะเกิดการรวมค่าแล้วหักล้างกันทำให้ไม่พบผลของการรีเซ็ต โดยกราฟทั้งสองแสดงค่า LBP ของการค้นหาด้วยรีรันและรีโพลิชันกับปัญหา BAYS29 ส่วนตาราง จ.8 คือ การทดลองรีโพลิชันและรีรันโดยทดสอบกับปัญหาการเดินทางของพนักงานขาย โดยใช้พารามิเตอร์ตามตาราง 5.3 แต่สำหรับการทดลองนี้ใช้จำนวนรอบการค้นหา 5000 รอบ โดยตารางแสดงค่า ARR ส่วนตาราง จ.9 คือการทดลองรีโพลิชันและรีรันโดยทดสอบกับปัญหาการเดินทางของพนักงานขาย โดยใช้พารามิเตอร์ตามตาราง 5.2 โดยตาราง จ.9 แสดงค่า MBF และ SR



รูปที่ จ.17 LBP ของ RERUN กับปัญหา BAYS29



รูปที่ จ.18 LBP ของ RN กับปัญหา BAYS29

ตารางที่ จ.8 ผลของ ARR ของปัญหาการเดินทางของพนักงานขายด้วย RERUN และ REPOSITION

ขั้นตอนวิธี	RERUN	RE-POSITION (RN)
ชื่อปัญหา	ARR	ARR
burma14	7.52	8.99
ulysses16	6.55	8.9
gr17	6.55	8.93
gr21	5.35	8.79
ulysses22	5.09	8.83
fri26	4.64	8.85
bayg29	4.09	8.49
bays29	4.14	8.63
ผลรวม	43.93	70.41

ตารางที่ จ.9 ผลการค้นหาคำตอบของ TSP ด้วย PPSO, RERUN และ REPOSITION

ขั้นตอนวิธี	PPSO		RE-RUN		RE-POSITION (RN)	
	MBF	SR	MBF	SR	MBF	SR
burma14	3327.68	64	3324.56	88	3324.04	92
ulysses16	6878.27	35	6869.72	58	6867.16	64
gr17	2093.47	73	2088.17	85	2086.93	87
gr21	2806	46	2776.02	59	2780.57	61
ulysses22	7115.74	14	7087.95	24	7082.26	28
fri26	975.55	1	965.39	4	966.95	2
bayg29	1764.02	0	1737.41	0	1728.29	1
bays29	2096.35	1	2083.84	0	2078.94	2
ผลรวม		234		318		337

จากกราฟ จ.17 และ จ.18 ช่วงแรกของการค้นหาเส้นกราฟจะลดลงแล้วคงที่ เมื่อเส้นกราฟคงที่เป็นระยะเวลาจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีรันหรือรีโพซิชั่น หลังจากดำเนินการรีรันหรือรีโพซิชั่นเส้นกราฟจะมีค่าเพิ่มขึ้นเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่ลงกว่าเดิม หลังจากนั้นจะดำเนินการค้นหาด้วย PPSO จึงทำให้เส้นกราฟลดลง ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้คำตอบนั้นดีขึ้น โดยในกรณีที่เริ่มต้นรีเซ็ตที่จุด A สังเกตได้จากกราฟที่จะลู่ลงไปมากกว่าก่อนการรีเซ็ต ทำให้รีรันหรือรีโพซิชั่นสามารถช่วยปรับปรุงคำตอบของ PPSO ให้ดีขึ้น แต่อย่างไรก็ตามโดยส่วนมากแล้วอนุภาคจะลู่กลับเข้ามาหา GBEST เดิม โดยที่ไม่พบตำแหน่งที่ดีกว่า GBEST เดิม โดยในกรณีที่เริ่มต้นรีเซ็ตที่จุด B สังเกตได้จากกราฟที่จะลู่ลงกลับมาเท่าเดิม

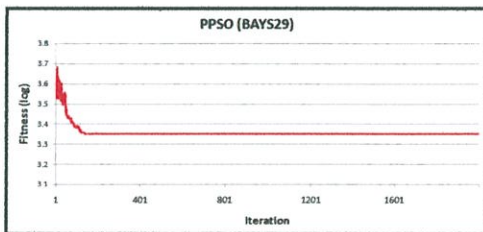
เมื่อสังเกตความสูงของเส้นกราฟหลังจากถูกรีเซ็ต จะพบว่า RN มีความสูงน้อยกว่า RERUN เนื่องจาก RN ได้เปลี่ยนแปลงตำแหน่งของอนุภาคเพียงบางส่วน จึงเหลือส่วนที่เป็นผลจากการค้นหาเดิมอยู่ทำให้ตำแหน่งของอนุภาคที่ดีที่สุดแย่ลงกว่าเดิมไม่มากนัก เส้นกราฟจึงสูงต่ำกว่าตอนเริ่มต้นค้นหา ในขณะที่ RERUN ทำการรีเซ็ตอนุภาคทั้งหมดทำให้ตำแหน่งของอนุภาคทั้งหมดถูกกำหนดใหม่

หมด เส้นกราฟจึงสูงเทียบเท่ากับตอนเริ่มต้นค้นหา ทำให้ขอบเขตการค้นหาของ RERUN มากกว่า RN ซึ่งทำให้ RERUN ต้องเสียเวลาในการลู่เข้าหากันมากกว่า RN

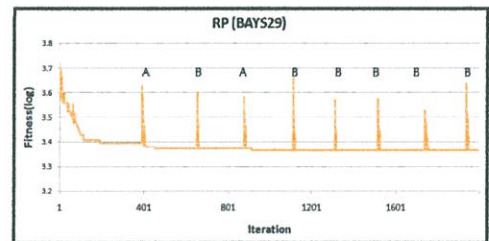
จากตารางผลการทดลอง จ.8 จำนวนรอบของการรีเซ็ตของ RN มีจำนวนมากกว่า RERUN หมายความว่ารีโพลิชชันมีโอกาสในการลองค้นหาใหม่มากกว่ารีรัน จากตารางผลการทดลอง จ.9 RN ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า PPSO และ RERUN นอกจากนี้ RN ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า RERUN โดยสรุปแล้วปัญหาการเดินทางของพนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือจำนวนรอบของการรีเซ็ตของ RN มีจำนวนมากกว่า RERUN และ RN ได้ผลลัพธ์จากการค้นหาโดยรวมดีกว่า PPSO และ RERUN

จ.6 การทดลองผลกระทบจากการรีเซ็ต PBEST และ GBEST ของรีโพลิชชัน

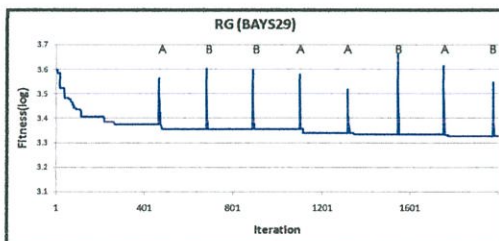
ในหัวข้อนี้จะพิจารณาถึงผลกระทบจากการรีเซ็ต GBEST และ PBEST ของรีโพลิชชันกับปัญหาการเดินทางของพนักงานขาย จากกราฟ จ.19 ถึง จ.22 คือการทดลองรีโพลิชชัน โดยใช้พารามิเตอร์ตามตาราง 5.3 แต่จำนวนรอบการค้นหาคำตอบ 2000 รอบ และจำนวนครั้งในการทดลองจะดำเนินการเพียงหนึ่งครั้ง เพราะแต่ละรอบของการทดลองการรีเซ็ตเกิดขึ้นไม่พร้อมกัน ถ้าเป็นค่าเฉลี่ยจะเกิดการรวมค่าแล้วหักล้างกันทำให้ไม่พบผลของการรีเซ็ต โดยกราฟทั้งสี่แสดงค่า LBP ของการค้นหาด้วย PPSO, RG, RP, RPG กับปัญหา BAYS29 สำหรับตาราง จ.10 คือการทดลอง PPSO, RG, RP, RN, RPG กับปัญหาการเดินทางของพนักงานขาย โดยใช้พารามิเตอร์ตามตาราง 5.3 โดยตาราง จ.10 แสดงค่า MBF และ SR



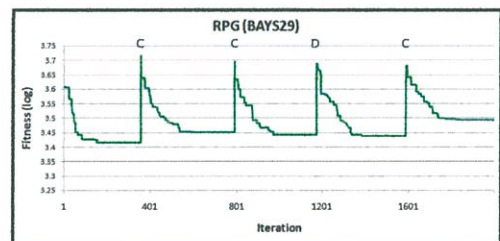
รูปที่ จ.19 LBP ของ PPSO กับปัญหา BAYS29



รูปที่ จ.20 LBP ของ RP กับปัญหา BAYS29



รูปที่ จ.21 LBP ของ RG กับปัญหา BAYS29



รูปที่ จ.22 LBP ของ RPG กับปัญหา BAYS29

ตารางที่ จ.10 ผลการค้นหาคำตอบของปัญหาการเดินทางของพนักงานขายด้วย PPSO, RG, RP, RN, RPG

ขั้นตอนวิธี	PPSO		RN		RP		RG		RPG	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
burma14	3327.55	65	3324.04	92	3323.26	98	3323.91	93	3323	100
ulysses16	6877.92	41	6867.16	64	6868.5	50	6867.78	62	6859	100
gr17	2096.3	69	2086.93	87	2088.86	67	2089.69	77	2085	100
gr21	2818.69	39	2780.57	61	2766.84	67	2773.96	56	2707	100
ulysses22	7122.33	6	7082.26	28	7091.68	11	7087.19	21	7014.95	95
fri26	973.43	1	966.95	2	964.63	6	964.92	3	943.44	58
bayg29	1753.5	0	1728.29	1	1715.72	4	1728.25	2	1635.58	2
bays29	2099.31	1	2078.94	2	2071.85	3	2079.39	1	2040.92	5
		222		337		306		315		560

จากกราฟ จ.19 ช่วงแรกเส้นกราฟจะมีค่าลดลง เนื่องจากผลจากการค้นหาด้วย PPSO และเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้นเส้นกราฟจะคงที่ตลอดจนสิ้นสุดการค้นหา จากกราฟ จ.18, จ.20, จ.21 ช่วงแรกของการค้นหาเส้นกราฟจะลดลงแล้วคงที่ เมื่อเส้นกราฟคงที่เป็นระยะเวลาจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีโพลิชชันและหลังจากดำเนินการรีโพลิชชัน เส้นกราฟจะมีค่าเพิ่มขึ้นเนื่องจากอนุภาคกระจายตัวออกทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่กว่าเดิม หลังจากนั้นจะดำเนินการค้นหาด้วย PPSO จึงทำให้เส้นกราฟลดลง ซึ่งระหว่างนี้อาจจะพบค่าที่ดีกว่า GBEST เดิมทำให้ค่าตอบนั้นดีขึ้น ในกรณีที่เริ่มต้นรีเซ็ตที่จุด A สังเกตได้จากกราฟที่จะลู่อลงไปมากกว่าก่อนการรีเซ็ต ทำให้รีรันหรือรีโพลิชชันสามารถช่วยปรับปรุงคำตอบของ PPSO ให้ดีขึ้น แต่อย่างไรก็ตามโดยส่วนมากแล้วอนุภาคจะลู่กลับเข้ามาหา GBEST เดิม โดยที่ไม่พบตำแหน่งที่ดีกว่า GBEST เดิม ในกรณีที่เริ่มต้นรีเซ็ตที่จุด B สังเกตได้จากกราฟที่จะลู่กลับมามีค่าเดิม

สำหรับจากกราฟ จ.22 ช่วงแรกของการค้นหาด้วย PPSO เส้นกราฟจะลดลงและเมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น เส้นกราฟจะคงที่ที่เป็นระยะเวลาจนสามารถสรุปได้ว่าในตอนนั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น จึงดำเนินการรีโพลิชชันหลังจากดำเนินการรีโพลิชชันเส้นกราฟจะมีค่าเพิ่มขึ้นเนื่องจากอนุภาคกระจายตัวออกจากกันทำให้ตำแหน่งของอนุภาคที่ดีที่สุดไปอยู่ในตำแหน่งที่แย่กว่าเดิม ซึ่งในบางครั้งอนุภาคก็ลู่เข้าค้นหาในบริเวณใหม่ ซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นที่จุด C เส้นกราฟหลังจากการรีเซ็ตมีค่าลดลงแล้วคงที่ในระดับที่ต่างจากเส้นกราฟก่อนการรีเซ็ต แต่ในบางครั้งอนุภาคก็ลู่เข้าค้นหาในบริเวณเดิม ซึ่งสังเกตได้จากกราฟที่จุดรีเซ็ตเริ่มต้นที่จุด D เส้นกราฟหลังจากการรีเซ็ตมีค่าลดลงแล้วคงที่ในระดับที่เท่าเดิม จากตารางผลการทดลอง จ.10 สำหรับ RPG ได้ผลลัพธ์โดยรวมดีที่สุดเมื่อเปรียบเทียบกับ PPSO, RP, RG, RN โดยสรุปแล้วปัญหาการเดินทางของพนักงานขายนั้นให้ผลลัพธ์การค้นหาเช่นเดียวกับฟังก์ชันมาตรฐาน คือ RPG ได้ผลลัพธ์โดยรวมดีที่สุดเมื่อเปรียบเทียบกับ PPSO, RP, RG, RN สำหรับพฤติกรรมการลู่เข้าหากันของ RPG กับการแก้ปัญหาการเดินทางของพนักงานขายที่ได้อธิบายไปนี้ มีลักษณะเหมือนกับพฤติกรรมการลู่เข้าหากันของ RPG กับฟังก์ชันมาตรฐาน

ภาคผนวก ฉ

รหัสจำลองของขั้นตอนวิธีการกลายพันธุ์และรีโพซิชัน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) ถ้าจำนวนรอบของการค้นหาค่าตอบหารด้วย RM ลงตัว
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PM > rand()$
ดำเนินการตามสมการ 4.2
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาค่าตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ฉ.1 รหัสจำลองของขั้นตอนวิธี FIM สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $j = 1$ ถึง RM
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = GBEST_d$
ถ้า $PM > rand()$
ดำเนินการตามสมการ 4.1
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาค่าตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ฉ.2 รหัสจำลองของขั้นตอนวิธี MGUG สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วงซ้ำจาก $j = 1$ ถึง RM
วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = PBEST_{id}$
ถ้า $PM > rand()$
ดำเนินการตามสมการ 4.1
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๓.3 รหัสจำลองของขั้นตอนวิธี MPUP สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วงซ้ำจาก $j = 1$ ถึง RM
วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_{id}$
ถ้า $PM > rand()$
ดำเนินการตามสมการ 4.1
ถ้า fitness ของ Tx ดีกว่า fitness ของอนุภาค i
 $x_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วงซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๓.4 รหัสจำลองของขั้นตอนวิธี MXUX สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $j = 1$ ถึง RM
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_{jd}$
ถ้า $PM > rand()$
ดำเนินการตามสมการ 4.1
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.5 รหัสจำลองของขั้นตอนวิธี MXUPG สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น (Threshold of Reposition, TR)
- 3) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 4) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 6) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 7) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 8) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
ทำการสุ่มตำแหน่งเริ่มต้นให้กับอนุภาคทุกตัว
- 9) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 3)

รูปที่ ๑.6 รหัสจำลองของขั้นตอนวิธี RERUN สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ดำเนินการตามสมการ 4.2
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๘.7 รหัสจำลองของขั้นตอนวิธี RN สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
รีเซ็ต GBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ดำเนินการตามสมการ 4.2
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๘.8 รหัสจำลองของขั้นตอนวิธี RG สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
รีเซ็ต PBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ดำเนินการตามสมการ 4.2
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.9 รหัสจำลองของขั้นตอนวิธี RP สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ 2.2
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ 2.3
- 10) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
รีเซ็ต GBEST, PBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ดำเนินการตามสมการ 4.2
- 11) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.10 รหัสจำลองของขั้นตอนวิธี RPG สำหรับปัญหาฟังก์ชันมาตรฐาน

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) ถ้าจำนวนรอบของการค้นหาคำตอบด้วย RM ลงตัว
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PM > rand()$
ถ้า $x_{id} = 1 : x_{id} = 0$
นอกจากนี้ : $x_{id} = 1$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.11 รหัสจำลองของขั้นตอนวิธี FIM สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $j = 1$ ถึง RM
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = GBEST_d$
ถ้า $PM > rand()$
ถ้า $Tx_d = 0 : Tx_d = 1$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.12 รหัสจำลองของขั้นตอนวิธี MGUG สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $j = 1$ ถึง RM
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = PBEST_{id}$
ถ้า $PM > rand()$
ถ้า $Tx_d = 0 : Tx_d = 1$
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.13 รหัสจำลองของขั้นตอนวิธี MPUP สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $j = 1$ ถึง RM
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_{id}$
ถ้า $PM > rand()$
ถ้า $Tx_d = 0 : Tx_d = 1$
ถ้า fitness ของ Tx ดีกว่า fitness ของอนุภาค i
 $x_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.14 รหัสจำลองของขั้นตอนวิธี MXUX สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) ววนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
 วนซ้ำจาก $j = 1$ ถึง RM
 วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 - $Tx_d = x_{id}$
 - ถ้า $PM > rand()$
 - ถ้า $Tx_d = 0 : Tx_d = 1$
 - ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
 - ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
 หยุดการดำเนินการค้นหาคำตอบ
 นอกจากนี้
 วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.15 รหัสจำลองของขั้นตอนวิธี MXUPG สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น (Threshold of Reposition, TR)
- 3) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 4) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
 เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
 เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 6) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 7) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 8) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
 ทำการสุ่มตำแหน่งเริ่มต้นให้กับอนุภาคทุกตัว
- 9) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
 หยุดการดำเนินการค้นหาคำตอบ
 นอกจากนี้
 วนซ้ำไปยังขั้นตอนที่ 3)

รูปที่ ๑.16 รหัสจำลองของขั้นตอนวิธี RERUN สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ถ้า $x_{id} = 1 : x_{id} = 0$
นอกจากนี้ : $x_{id} = 1$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.17 รหัสจำลองของขั้นตอนวิธี RN สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 9) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 10) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
รีเซ็ต GBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ถ้า $x_{id} = 1 : x_{id} = 0$
นอกจากนี้ : $x_{id} = 1$
- 11) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.18 รหัสจำลองของขั้นตอนวิธี RG สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
รีเซ็ต PBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ถ้า $x_{id} = 1 : x_{id} = 0$
นอกจากนี้ : $x_{id} = 1$
- 11) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.19 รหัสจำลองของขั้นตอนวิธี RP สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
รีเซ็ต PBEST GBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ถ้า $x_{id} = 1 : x_{id} = 0$
นอกจากนี้ : $x_{id} = 1$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.20 รหัสจำลองของขั้นตอนวิธี RPG สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ข.3
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ข.4 และ ข.5
- 9) วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วงซ้ำจาก $j = 1$ ถึง RM
วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_{id}$
ถ้า $PM > rand()$
ถ้า $Tx_d = 0 : Tx_d = 1$
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
รีเซ็ต PBEST GBEST
วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
ถ้า $x_{id} = 1 : x_{id} = 0$
นอกจากนี้ : $x_{id} = 1$
- 11) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วงซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ฉ.21 รหัสจำลองของขั้นตอนวิธี MRPSO สำหรับปัญหาถุงเป้หลายมิติ

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) ถ้าจำนวนรอบของการค้นหาค่าตอบหารด้วย RM ลงตัว
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PM > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โนด x_{id} กับ x_{ik}
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาค่าตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.22 รหัสจำลองของขั้นตอนวิธี FIM สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $j = 1$ ถึง RM
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = GBEST_d$
ถ้า $PM > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โนด Tx_d กับ Tx_k
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาค่าตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.23 รหัสจำลองของขั้นตอนวิธี MGUG สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วงซ้ำจาก $j = 1$ ถึง RM
วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = PBEST_{id}$
ถ้า $PM > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โมด Tx_d กับ Tx_k
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.24 รหัสจำลองของขั้นตอนวิธี MPUP สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วงซ้ำจาก $j = 1$ ถึง RM
วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_{jd}$
ถ้า $PM > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โนด Tx_d กับ Tx_k
ถ้า fitness ของ Tx ดีกว่า fitness ของอนุภาค i
 $x_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.25 รหัสจำลองของขั้นตอนวิธี MXUX สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดรอบของการกลายพันธุ์ (Round of Mutation, RM)
- 3) กำหนดความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation, PM)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
 วนซ้ำจาก $j = 1$ ถึง RM
 วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_d$
 ถ้า $PM > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
 สลับที่โนด Tx_d กับ Tx_k
 ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
 ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
 หยุดการดำเนินการค้นหาคำตอบ
 นอกจากนี้
 วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.26 รหัสจำลองของขั้นตอนวิธี MXUPG สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น (Threshold of Reposition, TR)
- 3) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 4) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
 เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
 เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 6) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 7) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 8) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
 ทำการสุ่มตำแหน่งเริ่มต้นให้กับอนุภาคทุกตัว
- 9) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
 หยุดการดำเนินการค้นหาคำตอบ
 นอกจากนี้
 วนซ้ำไปยังขั้นตอนที่ 3)

รูปที่ ๑.27 รหัสจำลองของขั้นตอนวิธี RERUN สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โนด x_{id} กับ x_{ik}
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.28 รหัสจำลองของขั้นตอนวิธี RN สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง \geq TR
รีเซ็ต GBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โนด x_{id} กับ x_{ik}
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๑.29 รหัสจำลองของขั้นตอนวิธี RG สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
รีเซ็ต PBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โน้ต x_{id} กับ x_{ik}
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๓.30 รหัสจำลองของขั้นตอนวิธี RP สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST
เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง
เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคตัวนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
รีเซ็ต GBEST, PBEST
วนซ้ำจาก $i = 1$ ถึง จำนวนประชากร
วนซ้ำจาก $d = 1$ ถึง จำนวนมิติ
ถ้า $PR > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
สลับที่โน้ต x_{id} กับ x_{ik}
- 10) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
หยุดการดำเนินการค้นหาคำตอบ
นอกจากนี้
วนซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๓.31 รหัสจำลองของขั้นตอนวิธี RPG สำหรับปัญหาการเดินทางของพนักงานขาย

- 1) ทำการสุ่มตำแหน่งและความเร็วเริ่มต้นให้กับอนุภาคแต่ละตัว (\vec{x}_i, \vec{v}_i)
- 2) กำหนดค่าที่กำหนดรอบของรีโพซิชั่น(Threshold of Reposition, TR)
- 3) กำหนดความน่าจะเป็นในการรีโพซิชั่น (Probability of Reposition, PR)
- 4) ประเมินค่าความเหมาะสมของอนุภาคแต่ละตัว (fitness)
- 5) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ GBEST เปลี่ยน GBEST เป็นอนุภาคนั้น ($GBEST = \vec{x}_i$)
- 6) ถ้า fitness ของอนุภาคใดๆ มีค่าดีกว่า fitness ของ PBEST ของตัวเอง เปลี่ยน PBEST ตัวนั้นเป็นอนุภาคนั้น ($PBEST_i = \vec{x}_i$)
- 7) ปรับปรุงความเร็วของแต่ละอนุภาคตามสมการ ค.1
- 8) ปรับปรุงตำแหน่งของแต่ละอนุภาคตามสมการ ค.2
- 9) วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
 วงซ้ำจาก $j = 1$ ถึง RM
 วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 $Tx_d = x_{id}$
 ถ้า $PM > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
 สลับที่โนด Tx_d กับ Tx_k
 ถ้า fitness ของ Tx ดีกว่า fitness ของ $PBEST_i$
 $PBEST_i = Tx$
 ถ้า fitness ของ Tx ดีกว่า fitness ของ GBEST
 $GBEST = Tx$
- 10) ถ้าจำนวนซ้ำของ GBEST อย่างต่อเนื่อง $\geq TR$
 รีเซ็ต PBEST GBEST
 วงซ้ำจาก $i = 1$ ถึง จำนวนประชากร
 วงซ้ำจาก $d = 1$ ถึง จำนวนมิติ
 ถ้า $PR > rand()$
 $k =$ ค่าที่สุ่มในช่วงของจำนวนมิติ โดยที่ $k \neq d$
 สลับที่โนด x_{id} กับ x_{ik}
- 11) ถ้า fitness ของ GBEST เท่ากับค่าที่กำหนด หรือ Iter เท่ากับ $Iter_{max}$
 หยุดการดำเนินการค้นหาคำตอบ
 นอกจากนี้
 วงซ้ำไปยังขั้นตอนที่ 4)

รูปที่ ๓.32 รหัสจำลองของขั้นตอนวิธี MRPSO สำหรับปัญหาการเดินทางของพนักงานขาย

ภาคผนวก ช

ผลงานวิจัยที่ได้รับการตีพิมพ์ (Full Paper)

1. Chiabwot Ratanavilisagul, Boontee Kruatrachue, “A modified particle swarm optimization with mutation and reposition”, International Journal of Innovative Computing, Information and Control (IJICIC), Volume 10, Number 6, December 2014, pp. 2127-2142

IJICIC

ISSN 1349-4198

Volume 10, Number 6, Dec 2014

***International Journal of Innovative
Computing, Information & Control***



Published by ICIC International
<http://www.ijicic.org>

A MODIFIED PARTICLE SWARM OPTIMIZATION WITH MUTATION AND REPOSITION

CHIABWOOT RATANAVILISAGUL AND BOONTEE KRUAETRACHUE

Department of Computer Engineering
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Rd., Ladkrabang, Bangkok 10520, Thailand
chaibwoot@hotmail.com

Received January 2014; revised May 2014

ABSTRACT. *The common problems of particle swarm optimization (PSO) are those of trapping in local optimum and premature convergence. This research paper aims to develop a solution to both problems by introducing mutation around particles and employing the reposition technique. The concurrent use of the introduced mutation and reposition has proved to solve both problems and enhanced the PSO performance; and thus is employed in this research. The proposed technique is termed MRPSO. MRPSO is tested on sixteen benchmark functions and the multidimensional knapsack problems (MKP). MRPSO yields the more satisfactory search results than the genetic algorithm (GA) and PSOs for the benchmark functions and the MKPs.*

Keywords: Particle swarm optimization, Binary particle swarm optimization, Mutation operator, Multidimensional knapsack problem, Genetic algorithm

1. Introduction. Kennedy and Eberhart [1,2] were the first to introduce the particle swarm optimization (PSO) in 1995 by observing the behaviors of animals, e.g., bird flocking and fish schooling. Their movements and communication mechanisms were thoroughly studied. In comparison with several other population-based stochastic optimization methods, such as the genetic algorithm (GA) [3,4] and the evolutionary programming (EP), PSO performs better in solving various optimization problems with fast and stable convergence rates [5-7].

The advantages of PSO [8,9] are its simplicity, rapid convergence, and few parameters to be adjusted. However, PSO has its own disadvantages of premature convergence to a local optimum and high chances of trapping in the local optimum [8,9]. To overcome both problems, researchers [9-27] increased search diversity in the population of PSO to prevent stagnation of the search in the local optimum by adding mutation operators in the PSO process.

This research has introduced mutation around individual particles (X) where the mutation values update the individual's best position (PBEST) and the best position found in the whole swarm (GBEST). In comparison with the standard PSO, this procedure increases search diversity without increasing the convergence speed; and is termed MXUPG. The experimental results of MXUPG on the benchmark test functions show MXUPG yields better solutions in the test functions than other mutation methods and the standard PSO.

The local optimum trapping is possible for both MXUPG and the standard PSO. This phenomenon leads to the stagnation of search in which the solution obtained at the

point of trapping is repeatedly produced irrespective of the length of search time. Consequently, certain research studies [28-31] added the reinitialize particles method (i.e., re-initialization) into PSO to improve the search solution. However, the reinitialize particles method requires a restart when trapping occurs. Therefore, an additional time is needed to re-converge in the new run. The particles reposition method (i.e., reposition) is proposed in this research in place of the reinitialize particles method to reduce the required re-convergence time. In addition, resetting both PBEST and GBEST yields the best results since the particles have more chances to converge in new areas after reposition. Hence, the particles reposition method with the resetting of both PBEST and GBEST (to be called RPG) is used to reduce the re-convergence time and increase the possible convergences in the new areas. In addition, the experimental results of RPG on the benchmark test functions yield better solutions in the test functions than other re-initialization methods and the standard PSO.

To improve the search performance, a novel PSO algorithm is created by combining MXUPG and RPG (to be called MRPSO). A set of benchmark test functions is used to compare the proposed MRPSO method with the standard PSO [1,2], an adaptive particle swarm optimization (APSO) [18], a hybrid particle swarm optimization which incorporates henon map mutation operation (HPSO) [19], the particle swarm optimization with reinitialize particles (PSOR) [28], and the floating point representation for genetic algorithm (FGA) [32]. The results show that the proposed MRPSO yields the best solutions in all test functions. Moreover, tests on the multidimensional knapsack problems (MKP) are performed with the proposed MRPSO, the binary representation for genetic algorithm (BGA) [32], the standard binary particle swarm optimization (BPSO) [33], the genotype-phenotype modified binary particle swarm optimization (GPMBPSO) [34] and the modified binary particle swarm optimization (MBPSO) [35]. The results are then compared. The comparison shows that the proposed MRPSO yields the best solutions in all instances of MKP.

The rest of this paper is organized as follows. Section 2 describes the standard PSO and BPSO. The performance analysis of the standard PSO and the PSO with mutation is detailed in Section 3. The proposed MRPSO method is discussed in Section 4 while Section 5 presents the experiment setup, the experiment results, and the detailed result analysis. Section 6 discusses certain applications of the proposed method. Section 7 is the conclusion.

2. Particle Swarm Optimization and Binary Particle Swarm Optimization.

2.1. Particle Swarm Optimization. In the standard PSO, each population member is called a "particle" with its own position and velocity. Each individual particle performs search in the search space according to its velocity, GBEST and PBEST. The standard PSO algorithm starts with randomizing particle positions and their respective velocities. The position evaluation of each particle is achieved using the objective function of the optimization problem. In a given iteration, each individual particle updates its position and velocity according to the expression below:

$$V'_{id} = \omega V_{id} + \eta_1 \text{rand}() (P_{id} - X_{id}) + \eta_2 \text{rand}() (P_{gd} - X_{id}) \quad (1)$$

$$X'_{id} = X_{id} + V'_{id} \quad (2)$$

where X'_{id} is the current positions of i particle and d dimension, X_{id} is the previous positions of i particle and d dimension, and V_{id} is the previous velocity of i particle and d dimension. V'_{id} is the current velocity of i particle and d dimension, P_{id} is PBEST of i particle and d dimension, and P_{gd} is GBEST of d dimension. η_1 and η_2 are acceleration

constants, $0 \leq \varpi < 1$ is an inertia weight, and $rand()$ generates random number from interval $[0, 1]$. A limit velocity is represented with V_{max} . Thus, if the calculated velocity of a particle exceeds V_{max} , it will be replaced with V_{max} .

2.2. Binary Particle Swarm Optimization. Kennedy and Eberhart [33] were the first to introduce the binary particle swarm optimization (BPSO) algorithm to allow the standard PSO to operate in the binary problem spaces. BPSO has since been adopted as a standard. In this algorithm, BPSO updates the velocity according to (3). The velocity is a set of real numbers and the particle position is a set of bits. Consequently, the velocity must be transformed into a set of probabilities using the sigmoid function as shown in (4). The principle of the position update is that the velocity dictates a probability that a position (bit) selects either zero or one; and the position update follows (5), where $sig(v'_{id})$ is the sigmoid function for transforming the velocity into the probability.

$$v'_{id} = v_{id} + \eta_1 rand()(P_{id} - X_{id}) + \eta_2 rand()(P_{gd} - X_{id}) \quad (3)$$

$$sig(v'_{id}) = \frac{1}{1 + e^{-v'_{id}}}, \quad e \approx 2.7182818 \quad (4)$$

$$X'_{id} = \begin{cases} 1 & \text{If } rand() < sig(v'_{id}) \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

3. The Performance Analysis of Standard PSO and PSO with Mutation.

3.1. Performance analysis of standard PSO. The search performance of the standard PSO is subject to types of optimization problems. For instance, if it is a unimodal problem which consists of one single global optimum point [36], the standard PSO performs well although the time required to search for the global optimum point is long. On the other hand, in a multimodal problem in which there are many local optimum points and one global optimum [36], the local optimum trapping frequently occurs in the standard PSO due to the position update [10], the effect of (1) [17], and the attraction of GBEST [23].

3.2. Performance analysis of PSO with mutation. This subsection discusses the performances after adding mutation operators in the PSO process; and the advantages and disadvantages.

The major advantage of mutation implementation is the increased population variability. The mutation could reduce the trapping of the standard PSO and yield the better search results. On the other hand, the disadvantage is the difficulty of adjusting the mutation parameters to any specific problems. In other words, if the parameters were under-mutated, the trapping could occur, making the mutation useless.

In the case of a non-compared mutation (NCM) operator [9-18] in which a mutation value from any given point is applied without comparing against the original value prior to mutation, if the parameters were over-mutated, the convergence of particles would be less likely. This induces a random search and the search result is poorer than that of the standard PSO.

In the case of a compared mutation (CM) operator [19-27] in which a mutation value from any given point is applied only if the mutation value is better than the original value prior to mutation, over-mutation of the parameters in a multimodal problem leads to a faster convergence speed than the standard PSO. This could result in premature convergence and a poorer search result than the standard PSO. In addition, CM is sometimes called the local search [23,24].

4. A Modified Particle Swarm Optimization with Mutation and Reposition.

4.1. The mutation around a point. To improve the performance of CM in multimodal problems, this research has proposed the novel algorithm that maintains the variability in the population without significant increase in the convergence speed when compared with the PSO. In applying the novel algorithm, each iteration mutation is introduced around particles. PBEST and GBEST are compared and updated with the resulting mutation values, rather than directly updating the particle positions with the mutation values. The update criteria are that if the resulting mutation values are better than the existing PBEST and GBEST, the PBEST and GBEST are replaced with the mutation values; and that if the resulting mutation values are poorer than the current PBEST and GBEST, both PBEST and GBEST are retained. The goal of this indirect update is to maintain the convergence speed of the standard PSO. The proposed novel algorithm is called the "mutation around particle position and update of PBEST and GBEST (MXUPG)". The MXUPG method is less likely to disrupt the convergence of particles since the update of particle positions is regulated by the standard PSO.

4.2. The particles reposition. As previously mentioned, the major shortcoming of the standard PSO is the frequent local optimum trapping. Fortunately, the trapping is readily detectable since GBEST is not updated for an indefinite period of time, as shown in Figure 3.

In the multimodal problem, when trapping is detected, the re-initialization method restarts another round of search until a maximum iteration is reached, rather than leaving the trapping problem unsolved. This allows the reinitialization particles to search for solutions in other areas and returns the better search solutions than the standard PSO.

However, the re-initialization process requires the initializing of all particles. In addition, a significant amount of time is needed for the particles to converge in each subsequent restart. The amount of time required for re-convergence in each restart reduces the number of possible reruns and subsequently the opportunity to search for better solutions in other areas. To quicken the convergence in each new run, this research has introduced the particles reposition method into the standard PSO.

The reposition is the use of NCM to allow for a vast distribution of particles prior to the solution search by the standard PSO. The reposition is different from the re-initialization in that it applies mutation to certain dimensions of the particles instead of re-initializing all values.

However, the inappropriate application of the re-initialization and reposition methods should be avoided since overuse could disturb the convergence of particles and results in a random search. On the other hand, the methods should be applied when local optimum trapping is detected and the trapping is long enough to guarantee that the trapping problem does occur.

When the reposition begins, particles change their positions while PBEST and GBEST could be either altered or unaltered. In the case of unaltered GBEST or unaltered PBEST or both unaltered before reset, the particles return from their new positions to the previous GBEST and PBEST. In the return to the previous GBEST, if the particles locate a better position than the previous GBEST, the better position is then occupied as a new GBEST. Typically, the positions along the return path are worse than the previous GBEST. Therefore, the swarm often returns to the previous GBEST and is re-trapped in the former area. This restricts the search area to the area around the previous GBEST.

To avoid the search area restriction in the reposition, both GBEST and PBEST should be reset. The reset allows particles to converge in new areas. The new areas could be either better or worse than the areas prior to reset since the process is random, as shown

by the restart from point A in Figure 3. However, resetting both GBEST and PBEST increases the chances of searching in better areas although there is a possibility that the swarm returns to the pre-reset area even with the resetting of both GBEST and PBEST, as shown by the restart from point B in Figure 3. The reposition method used together with the resetting of both PBEST and GBEST is called RPG.

The RPG method can solve the local optimum trapping when it occurs and reduce the required re-convergence time. Moreover, the method increases the chances of convergence in new areas after reposition.

4.3. A modified particle swarm optimization with mutation and reposition. MXUPG and RPG can be concurrently applied to improve the PSO performance. MXUPG increases the variability in the population to avoid the local optimum trapping and RPG solves the trapping problem when it occurs. Therefore, this research proposes integrating the two methods to improve the PSO performance. The use of the integrated methods (i.e., MXUPG and RPG) yields better results than the use of either one of the methods, as shown in Tables 2 and 3. The proposed integrated method is called MRPSO.

MRPSO begins searching using MXUPG and the search stops when an optimal solution is found. On the other hand, if a trapping occurs, RPG is activated to solve the trapping. When the trapping is solved, MXUPG resumes the search. RPG is reactivated if another trapping is detected until another optimal solution is found. The pseudo code of MRPSO is shown below:

```

Initial particles of each particle                                1
While (termination condition  $\neq$  true) do                       2
  Evaluate the fitness of each particle                         3
  If fitness of each particle is better than PBEST, update PBEST 4
  If fitness of each particle is better than GBEST, update GBEST 5
  Update each particle position according to (1) and (2)        6
  For  $i = 1$  to  $N$                                               (Mutation step) 7
    For  $j = 1$  to  $RM$                                            8
      For  $d = 1$  to  $D$                                            9
         $Tx_d = x_{id}$                                           10
        If  $PM > rand()$  then                                  11
          Apply (6)                                           12
        End If                                               13
      Next  $d$                                                   14
      If fitness of  $Tx$  is better than PBEST, update PBEST =  $Tx$  15
      If fitness of  $Tx$  is better than GBEST, update GBEST =  $Tx$  16
    Next  $j$                                                   17
  Next  $i$                                                     (End mutation step) 18
  If (times of GBEST consecutive unchanged)  $\geq TR$            (Reposition step) 19
    Reset GBEST, PBEST                                       20
    For  $i = 1$  to  $N$                                            21
      For  $d = 1$  to  $D$                                            22
        If  $PR > rand()$  then                                  23
          Apply (7)                                           24
        End If                                               25
      Next  $d$                                                   26
    Next  $i$                                                   27
  End If                                                    (End reposition step) 28
End while                                                    29

```

Equations (6) and (7) are mutation equations used in MRPSO. A positive operator is selected if the random number generated uniformly in the range $[0, 1]$ is less than 0.5 and

a negative operator otherwise.

$$Tx_d = Tx_d \pm (Tx_d \times rand()) \tag{6}$$

$$x_{id} = x_{id} \pm (x_{id} \times rand()) \tag{7}$$

where RM is the rounds of mutation. PM is the probability of mutation, PR is the probability of reposition, TR is the threshold of reposition, and Tx is the temporary particle. Tx_d is the temporary particle of d dimension, x_{id} is the positions of i particle of d dimension. N is the size of the population, and D is the dimension of the solution space.

5. Results and Discussions. To prove the efficiency of the proposed MRPSO, it is tested on sixteen benchmark functions [37,38], as listed in Table 1. Functions 1-9 are multimodal problems while the remaining Functions 10-16 are unimodal problems. The MRPSO results on the benchmark functions are compared with those of PSO, FGA, APSO, HPSO, PSOR, MXUPG, and RPG. The benchmark functions are representative of the floating-point representation. To prove the MRPSO ability in solving other representations, MKP has been added as a binary representation. This research selects the following MKPs from OR-Library [39], i.e., SENTO [40], WEING [41], WEISH [42], PB [43] and HP [43]. The MRPSO results on MKPs are compared with those of BPSO, BGA, GPMBPSO, MBPSO, MXUPG, and RPG.

TABLE 1. Details of benchmark test functions

Problem no.	Function name	Expression	Search space [X _{max} , X _{min}]	Objective function value	Dim	Attribute	V _{max}
1	Ackley	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \epsilon$	$x \in [-32.768, 32.768]^n$	0	50	Multimodal	32.768
2	Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	$x \in [-300, 300]^n$	0	50	Multimodal	300
3	Rastrigin	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$x \in [-5.12, 5.12]^n$	0	50	Multimodal	5.12
4	RosenBrock	$f(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$x \in [-2.048, 2.048]^n$	0	50	Multimodal	2.048
5	Schwefel	$f(x) = 418.9829 \times n + \sum_{i=1}^n \left(x_i \times \sin\left(\sqrt{ x_i }\right)\right)$	$x \in [-500, 500]^n$	0	50	Multimodal	500
6	Schaffer's F6	$f(x) = 0.5 + \frac{0.5 \sqrt{23+2\sqrt{5}}}{(1.6-0.5091 \times (x^2 - \rho))^2}$	$x \in [-100, 100]^n$	0	2	Multimodal	100
7	Step	$f(x) = \sum_{i=0}^{n-1} ((x_i + 0.5)^2)$	$x \in [-5.12, 5.12]^n$	0	50	Multimodal	5.12
8	Cosine Mixture	$f(x) = -0.1 \times \sum_{i=0}^n \cos(\pi x_i) + \sum_{i=0}^n x_i^2 + 0.1n$	$x \in [-1, 1]^n$	0	50	Multimodal	1
9	Exponential	$f(x) = -\exp(-0.5 \sum_{i=1}^n x_i^2) + 1$	$x \in [-1, 1]^n$	0	50	Multimodal	1
10	Spherical	$f(x) = \sum_{i=1}^n x_i^2$	$x \in [-5.12, 5.12]^n$	0	50	Unimodal	5.12
11	Parallel Ellipsoid	$f(x) = \sum_{i=1}^n (i \times x_i^2)$	$x \in [-5.12, 5.12]^n$	0	50	Unimodal	5.12
12	Multimod	$f(x) = \sum_{i=1}^n x_i \times \prod_{i=1}^n x_i $	$x \in [-10, 10]^n$	0	50	Unimodal	10
13	Rotated Ellipsoid	$f(x) = \sum_{i=1}^n (\sum_{j=1}^n x_j^2)$	$x \in [-65.536, 65.536]^n$	0	50	Unimodal	65.536
14	Zakharov	$f(x) = \sum_{i=1}^n x_i^2 + \left[\sum_{i=1}^n \frac{1}{2} x_i\right]^2 + \left[\sum_{i=1}^n \frac{1}{2} x_i\right]^4$	$x \in [-5.12, 5.12]^n$	0	50	Unimodal	5.12
15	Cigar	$f(x) = x_1^2 + 100000 \sum_{i=2}^n x_i^2$	$x \in [-10, 10]^n$	0	50	Unimodal	10
16	Brown 3	$f(x) = \sum_{i=0}^{n-1} \left[(x_i^2)^{2i+1} + (x_{i+1}^2)^{2i-1} \right]$	$x \in [-1, 1]^n$	0	50	Unimodal	1

This research uses a personal computer with Intel Core i7 3770 with a 2.4-GHz CPU and 8 GB RAM, and Visual C++ as the programming language. To guarantee fairness, the positions and velocity of particles are set identical in all the experiments. The measures of algorithm performance in the experiments are as follows.

The mean best fitness value (MBF): MBF is the mean of best fitness in the final iteration from all running and indicates the search efficiency of an algorithm. In the case of the experiments with the benchmark functions, all functions have a zero as the minimum point. In Table 1, an entry less than 10^{-324} is given a value of zero. The closer the MBF is to the zero point, the better the method is. In the case of MKP, where the best known values are given, the closer the MBF is to the best known value, the better the method is.

The success round (SR): SR is the number of running rounds that yields the optimum solution and shows the reliability of an algorithm. A good algorithm can be identified with a high SR. The higher the SR is, the better the algorithm is.

The logarithm mean fitness values of all particles (LMFP): LMFP is the logarithm mean of fitness of all particles in each iteration. For the benchmark functions, any algorithm that produces LMFP closer to the zero point is a good algorithm. The faster the LMFP is to entering the stabilizing stage, the faster the convergence speed is.

The logarithm of population average distance among points (LD(t)): $D(t)$ [44] is the distribution discrete degree between the particles in the population. $D(t)$ indicates the diversity of the population. In PSO, $D(t)$ can be calculated from (8). In MXUPG, $D(t)$ is summation of the results of (8) and (9).

$$D(t) = \frac{1}{M \times L} \sum_{i=1}^M \sqrt{\sum_{d=1}^D (x_{id}^t - \bar{p}_{id})^2} \quad (8)$$

$$D(t) = \frac{1}{M \times L \times RM} \sum_{r=1}^{RM} \sum_{i=1}^M \sqrt{\sum_{d=1}^D (mx_{idr}^t - \overline{mp}_{idr})^2} \quad (9)$$

where L is the diagonal length of search space, M is the size of the population, D is the dimension of the solution space, and x_{id}^t is the d dimension coordinate values of the i particle. \bar{p}_{id} is the average value of the d dimension coordinate values of all particles, mx_{idr}^t is the d dimension coordinate mutation values from the i particle in the j round of mutation, and \overline{mp}_{idr} is the average value of the d dimension coordinate values of mutation position from all particles in all rounds of mutation.

5.1. Experiment of the mutation around a point. This subsection discusses the property of the proposed MXUPG algorithm in relation to the standard PSO. The parameters of the benchmark test functions are presented in Table 1. The PSO parameters are as follows. η_1 and η_2 are 1.496180 and ϖ is 0.729844, as suggested by Bergh [30]. V_{\max} is shown in Table 1. PM and RM are 0.10 and 5 rounds, respectively. To guarantee fairness, the number of evaluations is set identical in all methods, resulting in 1200 particles in PSO and 200 particles in MXUPG. In Figures 1 and 2, each function has 50 runs. The MXUPG pseudo code is similar to the MRPSO pseudo code except that no reposition step is present in MXUPG.

As seen in Figures 1 and 2, LD(t) and LMFP are high during the initial search stage. They decrease and then stabilize toward the end. As shown in Figure 1, at the stabilizing stage, LD(t) of PSO is much less than LD(t) of MXUPG. This phenomenon indicates that the population variability of MXUPG is more than that of PSO. Figure 2 shows that LMFPs of PSO and MXUPG stabilize at approximately 1200 iterations. Thus, the convergence speeds of MXUPG and PSO are almost identical. It is possible to conclude that MXUPG is less likely to induce to premature convergence.

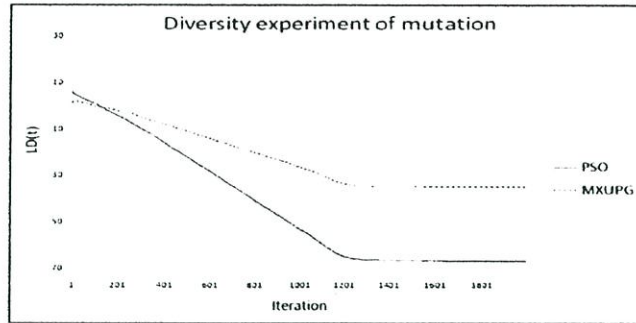


FIGURE 1. LD(t) of PSO and MXUPG of Ackley function

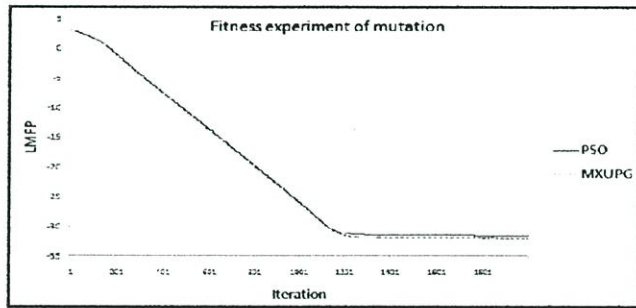


FIGURE 2. LMFPs of PSO and MXUPG of Ackley function

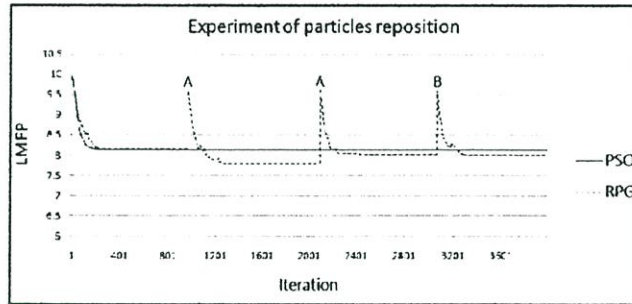


FIGURE 3. LMFPs of PSO and RPG of Schwefel function

5.2. **Experiment of particles reposition.** This subsection examines the property of the proposed RPG in relation to the standard PSO. In the reposition experiments, the parameters are set similarly to Subsection 5.1 except that the number of particles is 200 in all methods. In addition, TR and PR are set at 100 and 0.70, respectively. In Figure 3, this function runs only once since an average of the resetting results is difficult to detect.

The RPG pseudo code is similar to the MRPSO pseudo code except that the mutation step is not present in the RPG.

Figure 3 illustrates the local optimum trapping of PSO where LMFP initially decreases and then stabilizes. In the case of RPG, LMFP initially decreases and stabilizes until the local optimum trapping is detected. When the trapping occurs, the reposition is executed and rapidly increases LMFP. Afterward, the standard PSO starts searching for solutions and thus causes a decrease in LMFP.

5.3. Experiment of proposed algorithm on benchmark functions. This subsection investigates the performance of the proposed MRPSO on the benchmark functions listed in Table 1. The search results of MRPSO are compared with those of PSO, FGA, APSO, HPSO, RPG, PSOR and MXUPG. The PSO parameters, which are the underlying parameters of all algorithms, are identical to those in Subsection 5.1 except for the number of particles. The particles of PSO, FGA, APSO, HPSO, RPG, and PSOR are each 1200 particles while those of MRPSO and MXUPG are each 200 particles.

The non-PSO parameters are as follows. For APSO and HPSO, all parameters are identical to those in the original papers of APSO [18], HPSO [19] and PSOR [28], respectively. For FGA, the crossover probability and the mutation probability were set to 0.8 and 0.1, respectively, according to [45,46]. For RPG, MXUPG and MRPSO, the parameters are similar to those in Subsections 5.1 and 5.2. To guarantee fairness of the performance measurement, the mutation Equations (6) and (7) are applied to all methods. In Table 2, each function has 100 runs with a maximum iteration of 40000 iterations.

TABLE 2. Comparative results of PSO, FGA, APSO, HPSO, PSOR, RPG, MXUPG and MRPSO on the benchmark test functions

Techniques	PSO		FGA		APSO		HPSO		PSOR		MXUPG		RPG		MRPSO	
Problem no.	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
1	1.21E-14	0	7.10e-015	0	0.109254	0	6.04E-15	0	2.39E-14	0	7.11E-15	0	0	100	0	100
2	0.00669239	54	0	100	0.00447973	67	0	100	0.00334614	80	0	100	0	100	0	100
3	91.138	0	0	100	147.532	0	92.1927	0	60.9643	0	0	100	0	100	0	100
4	0	100	13.521	0	0	100	0	100	0	100	0	100	0	100	0	100
5	2395.47	0	2221.31	0	2545.11	0	2283.74	0	2530.5	0	0	100	1968.11	0	0	100
6	0	100	0	100	0	100	0	100	0	100	0	100	0	100	0	100
7	0	100	0	100	0.22	86	0	100	0.05	95	0	100	0	100	0	100
8	0.0709364	63	0	100	0.791897	0	0	100	0.0635472	18	0	100	0	100	0	100
9	1.24E-16	0	1.11e-016	0	2.02E-16	0	7.11E-17	36	2.69E-16	0	1.09E-16	2	0	100	0	100
10	0	100	0	100	0	100	0	100	6.43E-317	0	0	100	0	100	0	100
11	0	100	0	100	0	100	0	100	2.23E-279	0	0	100	0	100	0	100
12	0	100	0	100	2.11E-159	64	0	100	0	100	0	100	0	100	0	100
13	0	100	0	100	0	100	0	100	1.90E-287	0	0	100	0	100	0	100
14	1.10E-148	0	1.3256e-154	0	4.94066e-321	51	1.26E-160	0	0.353901	0	0	100	5.27E-147	0	0	100
15	0	100	0	100	0	100	0	100	2.53E-192	0	0	100	0	100	0	100
16	0	100	0	100	0	100	0	100	9.57E-306	0	0	100	0	100	0	100

As seen in Table 2, for the multimodal functions, RPG and MXUPG generally outperform APSO, FGA, HPSO, PSOR, and PSO. Hence, RPG and MXUPG can improve the PSO performance. RPG can solve the local optimum trapping when it occurs. In addition, RPG can increase the possibility of convergence in new areas after reposition and reduce the required re-convergence time. Hence, RPG produces better results than PSO in the multimodal problems. MXUPG increases variability without increase in the convergence speed when compared with the standard PSO. The search results of MXUPG are better than those of the standard PSO. For the unimodal functions, the search results of RPG are similar to those of PSO since the reposition is not executed. In some multimodal functions, MXUPG is unable to locate the optimum points due to local optimum trapping. The MBF and SR results indicate that MRPSO can locate the optimum points

of all functions in Table 1. MRPSO outperforms PSO, FGA, APSO, HPSO, PSOR, RPG and MXUPG in the floating-point representation problems. MRPSO is obtained from the combination of RPG and MXUPG. Thus, the application of MXUPG with RPG produces better search results than the use of either MXUPG or RPG.

PSO yields the better search results than FGA in some functions, such as the Rosenbrock function. On the other hand, the research results of FGA are better than those of PSO in some other functions, e.g., the Rastrigin function. This indicates that PSO and FGA are appropriate for solving different optimization problems. Table 2 shows that MRPSO could overcome the drawbacks of the standard PSO.

5.4. Experiment of proposed algorithm on multidimensional knapsack problem. The multidimensional knapsack problem (MKP) is a widely researched discrete programming problem [47] and belongs to the group of NP-hard combinatorial optimization problems [47-49]. The MKP formula can be applied to various business requirements, such as the project selection problem, the capital budgeting problem, the cutting stock problem, the cargo loading problem, the production planning problem, the scheduling problem, the distributed processor problem, and the database allocation problem [49,50]. Hence, an algorithm that can solve an MKP would be able to solve every optimization problem with the same MKP formula and has many uses [50].

MKP consists of m knapsacks with maximum capacity C_j ($j = 1, \dots, m$) for each knapsack and a set of n objects, where each object i ($i = 1, \dots, n$) has a price (P_i) and a weight (w_{ij}). The goal of MKP is to select a subset of objects that maximizes the total price without exceeding the total capacity of all knapsacks. MKP can be expressed as

$$\text{Maximize } \sum_{i=1}^n p_i x_i \quad x_i \in \{0, 1\} \quad (10)$$

$$\text{Object } \sum_{i=1}^n w_{ij} x_i \leq C_j \quad x_i \in \{0, 1\} \quad (11)$$

This subsection investigates the performance of the proposed MRPSO on MKP. The search results of MRPSO are compared with those of BPSO, GPMBPSO, MBPSO, MXUPG, RPG and BGA. The application of MRPSO pseudo code to MKP requires changes to some parts of the MRPSO pseudo code. The changes and reasons are detailed as the following: a dimension (d) or bit, which represents an object, selects 1 if the object is selected and 0 otherwise. The standard PSO is replaced with BPSO, and if a bit changes from 0 to 1, BPSO checks the capacity of all knapsacks. If the capacity of all knapsacks can handle the weight of this object, the bit is allowed to change from 0 to 1; otherwise, the bit remains unchanged. The capacity check is to guarantee that the search does not exceed the solution space boundaries. The mutation step of the MRPSO pseudo code is to replace Line 12 (6) with $T_{rd} = 1$. The reposition step is edited by replacing Line 24 (7) with a flipped bit of x_{id} . In addition, a velocity reset is required since the particle velocity influences the particle bit change [51]. Unless the velocity is reset, the particle bits would return to the pre-reposition state and thus make the reposition useless.

The BPSO parameters are as follows: η_1 and η_2 are both set at 2, and V_{\max} is set at 4, as recommended by Deep [35]. On the other hand, the non-BPSO parameters are as follows: the parameters of GPMBPSO and MBPSO are identical to those in the original papers of MBPSO [35]. In RPG, MXUPG and MRPSO, PM is 0.05, RM is 1 round, TR is 30 iterations, and PR is 0.3. For BGA, the crossover probability and the mutation probability were set at 0.8 and 0.1, respectively, according to [45,46]. To guarantee fairness, the number of evaluations is set identical in all methods. This results

in 1000 particles each in BPSO, BGA, GPMBPSO, MBPSO and RPG; and 500 particles each in MRPSO and MXUPG.

In Table 3, each instance has 100 runs with a maximum iteration of 5000 iterations. In the same table, the results of MBF and sums of SR show that MRPSO can locate the optimum points of all instances. MRPSO outperforms BPSO, BGA, MBPSO, GPMBPSO,

TABLE 3. Comparative results of BPSO, BGA, GPMBPSO, MBPSO, MXUPG, RPG and MRPSO on MKP

Techniques		BPSO		BGA		GPMBPSO		MBPSO		MXUPG		RPG		MRPSO	
Problem	Best known	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
Sento1	7772	7771.3	95	7772	100	7772	100	7770.88	98	7771.72	98	7771.58	97	7772	100
Sento2	8722	8718.91	67	8722	100	8722	100	8719.97	74	8719.64	76	8720.61	79	8722	100
Weing1	141278	141278	100	141278	100	141278	100	141278	100	141278	100	141278	100	141278	100
Weing2	130883	130883	100	130883	100	130883	100	130883	100	130883	100	130883	100	130883	100
Weing3	95677	95660.6	96	95677	100	95676	98	95661.6	98	95676.5	99	95677	100	95677	100
Weing4	119337	119337	100	119337	100	119337	100	119337	100	119337	100	119337	100	119337	100
Weing5	98796	98796	100	98796	100	98796	100	98796	100	98796	100	98796	100	98796	100
Weing6	130623	130623	100	130623	100	130623	100	130623	100	130623	100	130623	100	130623	100
Weing7	1095450	1095340	6	1094150	0	1095380	0	1095370	27	1095300	8	1095410	90	1095450	100
Weing8	624319	620857	11	624319	100	620591	0	621966	11	624319	100	622038	32	624319	100
Weish01	4554	4554	100	4554	100	4554	100	4554	100	4554	100	4554	100	4554	100
Weish02	4536	4535.9	98	4536	100	4536	100	4536	100	4536	100	4535.9	98	4536	100
Weish03	4115	4115	100	4115	100	4115	100	4115	100	4115	100	4115	100	4115	100
Weish04	4561	4561	100	4561	100	4561	100	4561	100	4561	100	4561	100	4561	100
Weish05	4514	4514	100	4514	100	4514	100	4514	100	4514	100	4514	100	4514	100
Weish06	5557	5556.02	91	5557	100	5549.72	11	5556.46	97	5556.87	99	5556.87	99	5557	100
Weish07	5567	5567	100	5567	100	5567	100	5567	100	5567	100	5567	100	5567	100
Weish08	5605	5605	100	5605	100	5605	100	5605	100	5605	100	5605	100	5605	100
Weish09	5246	5246	100	5246	100	5246	100	5246	100	5246	100	5246	100	5246	100
Weish10	6339	6339	100	6339	100	6339	100	6339	100	6339	100	6339	100	6339	100
Weish11	5643	5643	100	5643	100	5643	100	5643	100	5643	100	5643	100	5643	100
Weish12	6339	6339	100	6339	100	6339	100	6339	100	6339	100	6339	100	6339	100
Weish13	6159	6159	100	6159	100	6159	100	6159	100	6159	100	6159	100	6159	100
Weish14	6954	6954	100	6944.08	68	6954	100	6954	100	6954	100	6954	100	6954	100
Weish15	7486	7486	100	7486	100	7486	100	7486	100	7486	100	7486	100	7486	100
Weish16	7289	7288.94	97	7289	100	7289	100	7288.78	97	7288.96	98	7289	100	7289	100
Weish17	8633	8633	100	8633	100	8633	100	8633	100	8633	100	8633	100	8633	100
Weish18	9580	9579.26	91	9578.16	76	9580	100	9579.85	99	9580	100	9580	100	9580	100
Weish19	7698	7697.71	98	7664.08	8	7695.14	78	7697.61	97	7698	100	7698	100	7698	100
Weish20	9450	9450	100	9449.4	88	9450	100	9450	100	9450	100	9450	100	9450	100
Weish21	9074	9074	100	9072.95	85	9074	100	9073.76	99	9074	100	9074	100	9074	100
Weish22	8947	8940.88	66	8922.2	5	8930.44	8	8940.7	65	8943.58	81	8945.38	91	8947	100
Weish23	8344	8343.79	93	8306.55	0	8343.94	98	8343.76	96	8343.94	98	8344	100	8344	100
Weish24	10220	10219.8	99	10215.6	80	10220	100	10220	100	10220	100	10220	100	10220	100
Weish25	9939	9935.32	78	9918.95	20	9939	100	9937.56	91	9938.01	93	9938.84	99	9939	100
Weish26	9584	9575.93	69	9550.9	21	9584	100	9576.96	78	9578.24	82	9582.72	96	9584	100
Weish27	9819	9819	100	9798.15	33	9819	100	9819	100	9819	100	9819	100	9819	100
Weish28	9492	9492	100	9459	20	9492	100	9491.77	99	9492	100	9492	100	9492	100
Weish29	9410	9410	100	9381.85	39	9410	100	9409.52	98	9410	100	9410	100	9410	100
Weish30	11191	11190.6	90	11131.5	0	11191	100	11190.8	94	11190.8	94	11191	99	11191	100
PB1	3090	3090	100	3090	100	3090	100	3090	100	3090	100	3090	100	3090	100
PB2	3186	3185.28	96	3113.92	14	3186	100	3184.2	93	3185.82	99	3185.82	99	3186	100
PB4	95168	95168	100	95168	100	95168	100	95168	100	95168	100	95168	100	95168	100
PB5	2139	2137.98	91	2129.36	50	2137.64	96	2139	100	2139	100	2138.83	99	2139	100
PB6	776	776	100	776	100	776	100	776	100	776	100	776	100	776	100
PB7	1035	1034.96	96	1035	100	1035	100	1034.8	98	1034.91	99	1035	100	1035	100
HP1	3418	3416.18	92	3418	100	3404	0	3416.61	95	3418	100	3417.58	97	3418	100
HP2	3186	3186	100	3106.48	4	3186	100	3185.82	99	3186	100	3185.82	99	3186	100
Sum of SRs			4432		3611		4322		4506		4624		4674		4800

RPG and MXUPG in terms of reliability and solution quality of the binary representation problems. This is shown by MBFs and the sums of SR, both of which are higher than other algorithms. In addition, Table 3 reaffirms the fact that the use of MXUPG with RPG produces better search results than the use of either MXUPG or RPG.

As shown in Table 3, the BPSO search results are better than those of BGA in some instances, such as Weish23 and Weish30. On the other hand, BGA gives better search results than BPSO in some other instances, e.g., Sento2 and Weing8. The phenomena confirm that both algorithms are capable of solving different optimization problems. In addition, the table shows that MRPSO could overcome the PSO drawbacks.

6. Applications of the Proposed Algorithm. The PSO and GA algorithms belong to the evolutionary algorithm (EA) group [52] and have several similar useful properties to EA [52]. This research demonstrates the practical use of the proposed MRPSO algorithm by applying it to the optimization problems and presents the benefits of EA.

The optimization problems are found in various business problems [53], such as the project selection problem, the capital budgeting problem, the cutting stock problem, the cargo loading problem, the production planning problem, the scheduling problem, the distributed processor problem, the database allocation problem, and the knapsack problem (KP). In addition, they are found in different logistic operations, e.g., the logistics and transportation problem, the shortest path problem, the maximum flow problem, the generalized assignment problem, the traveling salesman problem (TSP), and the vehicle routing problem. The optimization problems are also discovered in several other problem types, e.g., the blending problem and the crop planning problem.

In the case of optimization problems with small solution space, e.g., KP which consists of 5 items, the brute force method can locate the optimum point because the search is carried out with all feasible solutions (i.e., 32 solutions). On the other hand, in the case of optimization problems with very large solution space, such as KP with 500 items, the total number of feasible solutions is 2^{500} . Therefore, the brute force method is unsuitable since it requires a very long time for the method to locate the optimum point. In this case, other methods (e.g., PSO and GA) should be adopted to solve the optimization problems with very large solution space.

One major benefit of PSO is its applicability to the optimization problems with very large solution space. The basic principle of PSO is that it would not search for all feasible solutions but would focus on the areas that produce good results. This narrows down the scope of solution search. Therefore, PSO can search for the solutions very efficiently since it is typical to locate better solutions around the areas that previously produced good solutions.

A problem-specific algorithm produces a better optimization performance than an EA (e.g., PSO and GA), but the problem-specific algorithm is only applicable to the type of the optimization problem for which it was designed [36]; for example, the Lin-Kernighan heuristics can solve TSP efficiently but cannot be applied to MKP. On the other hand, the repair operator algorithm [54] can solve MKP efficiently but cannot be applied to TSP. In addition, implementing the problem-specific algorithm requires the knowledge of the method to solve that specific optimization problem. Hence, the implementation of the problem-specific algorithm is complicated and is limited to only one specific problem. In contrast, implementing PSO is not subject to the optimization problems. In other words, implementing PSO requires adjusting certain parameters to fit the problems [36], such as amount of generation, amount of population, and constraint of the optimization problem.

Another main benefit of PSO is its ease of implementation. In addition, after the implementation, PSO can be reused with other optimization problems.

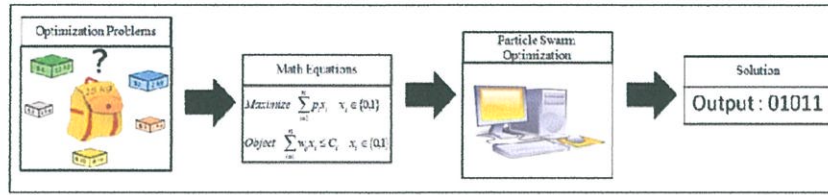


FIGURE 4. A practical application of PSO

Figure 4 illustrates a practical application of PSO. The optimization problems are converted into mathematical equations. If the equations can be solved, the optimization problems would be solved at the same time. Then, PSO searches for the solutions of the problems.

Many optimization problems contain local optimum in solution space [36], so if an algorithm were poorly designed, the local optimum trapping incidence would occur. If the trapping happened, the best solution would not be found. However, if there were no trapping, the best solution would be located quickly.

One main drawback of the standard PSO is a tendency of the local optimum trapping to occur. Therefore, the aim of this research is to minimize the local optimum trapping of PSO. The improved performance of PSO is achieved by combining the mutation technique with the reposition technique.

The experimental outcomes in Subsections 5.3 and 5.4 confirm that the proposed MRPSO method can improve the efficiency and effectiveness of PSO and that the best results are achieved with the proposed MRPSO when compared with the other algorithms. The proposed MRPSO has the highest efficiency because it has the same number of evaluations as the other algorithms but produces the best solutions. In terms of effectiveness, the proposed MRPSO achieves the highest effectiveness since it minimizes the local optimum trapping of PSO. Thus, the MRPSO algorithm can locate every global optimum point in the experiments, whereas PSO fails to locate the global optimum points in several functions and instances.

In practice, the proposed MRPSO is applicable to solving the optimization problems. In addition, it has been proved that the proposed MRPSO solves the optimization problems better than the standard PSO. Many previous research papers studied hybrid algorithms, such as that which combines PSO with the repair operator algorithm to solve MKP [54], and that which combines the Lin-Kernighan heuristics with PSO to solve TSP [55]. The hybrid algorithms improve the solution search performance. In the hybrid algorithms, if PSO were replaced with the proposed MRPSO, the MRPSO hybrid algorithm would produce a better result than the PSO hybrid algorithm.

7. Conclusion. This research paper has proposed MRPSO derived from the concurrent application of MXUPG and RPG to solving the optimization problems. MXUPG increases the population diversity without increasing the convergence speed relative to the standard PSO. This reduces the occurrence of local optimal trapping. However, since some trappings are inevitable with the application of only MXUPG, RPG is therefore introduced and used in combination. The reason is that RPG increases the chances of finding better solutions when the trappings occur.

Thus, the concurrent application of MXUPG and RPG is proposed to improve the PSO performance. On the benchmark functions, the MRPSO, PSO, FGA, APSO, PSOR and

HPSO are tested and the results are compared. In addition, the MRPSO, BGA, BPSO, GPMBPSO and MBPSO are tested on MKP and the results are compared. The results indicate that the proposed MRPSO outperforms PSO, FGA, APSO, HPSO, PSOR, BGA, BPSO, MBPSO, and GPMBPSO with regard to the reliability and quality of solutions in all the experiments. The proposed MRPSO is able to find the optimal points.

REFERENCES

- [1] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, Perth, Western Australia, vol.4, pp.1942-1947, 1995.
- [2] J. Kennedy and R. Eberhart, A new optimizer using particle swarm theory, *The 6th International Symposium on Micro Machine and Human Science*, pp.39-43, 1995.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1979.
- [4] D. E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [5] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas and M. N. Vrahatis, Objective function stretching to alleviate convergence to local minima, *Nonlinear Analysis, Theory, Methods and Applications*, vol.47, pp.3419-3424, 2003.
- [6] R. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, *Proc. of the 7th Annual Conference on Evolutionary Programming*, pp.611-619, 1998.
- [7] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Mateo, 2001.
- [8] B. Qinghai, Analysis of particle swarm optimization algorithm, *Computer and Information Science*, vol.3, no.1, pp.180-184, 2010.
- [9] R. Chiabwoot and K. Boontee, Mutation period calculation for particle swarm optimization, *The 1st International Symposium on Technology for Sustainability*, pp.213-216, 2011.
- [10] M. Lin and Z. Hua, Improved PSO algorithm with adaptive inertia weight and mutation, *World Congress on Computer Science and Information Engineering*, pp.627-625, 2009.
- [11] N. Hiquishi and H. Iba, Particle swarm optimization with Gaussian mutation, *IEEE Conf. Swarm Intelligence Symposium (SIS)*, pp.72-79, 2003.
- [12] R. Thangaraj, M. Pant and A. Abraham, A new diversity guided particle swarm optimization with mutation, *World Congress on Nature and Biologically Inspired Computing*, pp.294-299, 2009.
- [13] P. S. Andrews, An investigation into mutation operators for particle swarm optimization, *IEEE Congress on Evolutionary Computation*, pp.1044-1051, 2006.
- [14] A. Stacey, M. Jaucic and I. Grundy, Particle swarm optimization with mutation, *Proc. of IEEE International Conference on Evolutionary Computation*, vol.2, pp.1425-1430, 2003.
- [15] J. Wei, L. Guangbin and L. Dong, Elite particle swarm optimization with mutation, *Asia Simulation Conference 7th Intl. Conf. on Sys. Simulation and Scientific Computing*, pp.800-803, 2008.
- [16] J. Liu, J. Sun and W. B. Xu, Quantum-behaved particle swarm optimization with mutation operator, *Proc. of the 17th IEEE International Conference on Tools with Artificial Intelligence*, pp.237-240, 2005.
- [17] H. Choi, O. Shunichi, Y. Kazuho and O. Hiroaki, Improvement of particle swarm optimization application of the mutation concept for the escape from local minima, *The 8th International Conference on Supply Chain Management and Information Systems*, pp.1-5, 2010.
- [18] A. ALFI, PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems, *Acta Automatica Sinica*, vol.37, no.5, pp.541-549, 2011.
- [19] H. Gao and W. Xu, Particle swarm algorithm with hybrid mutation strategy, *Applied Soft Computing*, pp.5129-5142, 2011.
- [20] J. Tang and X. Zhao, Particle swarm optimization with adaptive mutation, *WASE International Conference on Information Engineering*, pp.234-237, 2009.
- [21] X. Wu and M. Zhong, Particle swarm optimization based on power mutation, *ISECS International Colloquium on Computing, Communication, Control, and Management*, pp.464-467, 2009.
- [22] C. Li, S. Yang and I. A. Korejo, An adaptive mutation operator for particle swarm, *Proc. of the UK Workshop on Computational Intelligence*, pp.165-170, 2008.
- [23] J. Tang and X. Zhao, A hybrid particle swarm optimization with adaptive local search, *Journal of Networks*, pp.411-418, 2010.
- [24] J. Tang and X. Zhao, Particle swarm optimization using adaptive local search, *International Conference on Future BioMedical Information Engineering*, pp.300-303, 2009.

- [25] M. Pant, R. Thangaraj and A. Abraham, Particle swarm optimization using adaptive mutation, *Proc. of the 19th International Conference on Database and Expert Systems Application*, pp.519-523, 2008.
- [26] L. Chen, Particle swarm optimization with a novel mutation operator, *International Conference on Mechatronic Science, Electric Engineering and Computer*, pp.970-973, 2011.
- [27] M. S. Darweesh, M. M. Ghoneimy and H. A. Kamal, A new fixed channel assignment algorithm using adaptive mutation particle swarm optimization, *Innovations on Communication Theory*, pp.1-5, 2012.
- [28] X. Liu, Q. Wang, H. Liu and L. Li, Particle swarm optimization with dynamic inertia weight and mutation, *The 3rd International Conference on Genetic and Evolutionary Computing*, pp.620-623, 2009.
- [29] Y. Gao and Y. Duan, A new particle swarm optimization algorithm with adaptive mutation operator, *The 2nd International Conference on Information and Computing Science*, pp.58-61, 2009.
- [30] F. van den Bergh, *An Analysis of Particle Swarm Optimizers*, Ph. D. Thesis, University of Pretoria, 2001.
- [31] L. Ning, S. Debao, C. Yigang and Z. Tong, Particle swarm optimization with mutation operator, *Computer Engineering and Applications*, vol.17, pp.12-14, 2004.
- [32] C. Z. Janikow and Z. Michalewicz, An experimental comparison of binary and floating point representation in genetic algorithm, *Proc. of the 4th International Conference on Genetic Algorithms*, pp.31-36, 1991.
- [33] J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm algorithm, *Proc. of the World Multiconference on Systemics, Cybernetics, Informatics*, pp.4104-4109, 1997.
- [34] S. Lee, S. Soak, S. Oh, W. Pedrycz and M. Jeon, Modified binary particle swarm optimization, *Progress in Natural Science*, pp.1161-1166, 2008.
- [35] J. C. Bausal and K. Deep, A modified binary particle swarm optimization for knapsack problems, *Applied Mathematics and Computation*, pp.11042-11061, 2012.
- [36] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, Berlin, 2003.
- [37] M. Marcin and S. Czeslwo, *Test Functions for Optimization Needs*, 2005.
- [38] P. A. Ernesto, *Multivariate Test Functions Library in C for Unconstrained Global Optimization*, Department of Mathematics U.P. Diliman, 2005.
- [39] J. E. Beasley, *ORLib - Operations Research Library*, <http://brunel.ac.uk/~mastjib/jeb/orlib/files/mknaps2.txt>.
- [40] S. Senyu and Y. Toyoda, An approach to linear programming with 0-1 variables, *Management Science*, vol.15, pp.196-207, 1967.
- [41] H. M. Weingartner and D. N. Ness, Methods for the solution of the multidimensional 0/1 knapsack problem, *Operations Research*, vol.15, pp.83-103, 1967.
- [42] W. Shih, A branch and bound method for the multicoustraint zero-one knapsack problem, *Journal of Operations Research Society*, vol.30, pp.369-378, 1979.
- [43] A. Freville and G. Plateau, Hard 0-1 multiknapsack test problems for size reduction methods, *Investigation Operativa*, vol.1, pp.251-270, 1990.
- [44] T. Krink, J. S. Vesterstrom and J. Riget, Particle swarm optimization with spatial particle extension, *Proc. of IEEE Congress on Evolutionary Computation*, Honolulu, HI, USA, pp.1474-1479, 2002.
- [45] J. J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, pp.122-128, 1986.
- [46] J. D. Schaffer, R. A. Caruana, L. J. Eshelman and R. Das, Study of control parameters affecting online performance of genetic algorithms for function optimization, *Proc. of the 3rd International Conference on Genetic Algorithms*, pp.51-60, 1989.
- [47] S. Martello and P. Toth, Bin-packing problem in knapsack problems, *Algorithms and Computer Implementations*, pp.221-245, 1990.
- [48] Y. Zhou, Z. Kuang and J. Wang, A chaotic neural network combined heuristic strategy for multidimensional knapsack problem, *ISICA*, pp.715-722, 2008.
- [49] P. C. Chu and J. E. Beasley, A genetic algorithm for the multidimensional knapsack problem, *Journal of Heuristics*, vol.4, pp.63-86, 1998.
- [50] J. Deane and A. Agarwal, Neural, genetic, and neurogenetic approaches for solving the 0-1 multidimensional knapsack problem, *International Journal of Management and Information Systems*, vol.17, pp.43-54, 2013.
- [51] M. Khaesar, M. Teshnehlab and M. Shoorehdeli, A novel binary particle swarm optimization, *Proc. of the 15th Mediterranean Conference on Control and Automation*, Greece, Athens, pp.1-6, 2007.

- [52] E. Elbeltagia, T. Hegazy and D. Griersonb, Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, pp.43-53, 2005.
- [53] R. A. Sarker and C. S. Newton, *Optimization Modelling: A Practical Approach*, CRC Press, Taylor and Francis Group, 2008.
- [54] M. Kong and P. Tian, Apply the particle swarm optimization to the multidimensional knapsack problem, *Artificial Intelligence and Soft Computing*, vol.4029, pp.1140-1149, 2006.
- [55] W. Pang, K. Wang, C. Zhou, L. Dong, M. Liu, H. Zhang and J. Wang, Modified particle swarm optimization based on space transformation for solving traveling salesman problem. *Proc. of the 3rd International Conference on Machine Learning and Cybernetics*, pp.2342-2346, 2004.

ผลงานวิจัยที่ได้รับการตีพิมพ์ (Short Paper)

2. Chiabwoot Ratanavilisagul, Boontee Kruatrachue, “A modified particle swarm optimization with dynamic mutation period”, Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 14-17 May 2014, pp. 1-6



ECTI-CON 2014

Nakhon Ratchasima, Thailand May 14-17, 2014

ECTI-CON 2014 is the eleventh annual international conference organized by Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association, Thailand. The conference aims to provide an international platform to present technological advances, launch new ideas and showcase research work in the field of electrical engineering, electronics, computer, telecommunications and information technology. Accepted papers will be published in the Proceedings of ECTI-CON 2014 and will be available via IEEE Xplore. Acceptance will be based on quality, relevance and originality.



Committee ECTI - CON 2014
Advisory Committee
 Monai Krairiksh (KMILT)
 Prabhas Chongstitvatana (CU)
 Rangsan Wongsan (SUT)



Steering Committee
 Tomoaki Sato (Hiroaki U, Japan)
 Yoshihiro Matsui (TNCT, Japan)
 Kou Yamada (Gunma U, Japan)
 Prayoot Akkaraekthalin (KMUTNB)
 Naruemon Wattananopongsakorn (KMUTT)
 Tupim Angkaew (CU)
 Kosin Chamnongthai (KMUTT)



General Chairs
 Prayoot Akkaraekthalin (KMUTNB)
 Chutima Prommak (SUT)



General Co-Chair
 Somsak Vanit-Anunchai (SUT)



General Secretary
 Wipawee Hattagarn (SUT)



Special session Chairs
 Yoshihiro Matsui (TNCT, Japan)
 Thanatchai Kulkorawanichpong (SUT)



Exhibition and Sponsor Chairs
 Rangsan Tongta (SUT)
 Chanchai Thongsopa (SUT)
 Prayoth Kumsawat (SUT)

Publicity Chairs
 Kou Yamada (Gunma U, Japan)
 Kongpol Areerak (SUT)
 Kongpan Areerak (SUT)

TPC Chair
 David Banjerdpongchai (CU)

TPC Co-chairs
 Kamol Kaemarungsi (NECTEC)
 Boon Hee Soong (NTU, Singapore)

International Technical Committee
 Liew Soung Yue (UTAR, Malaysia)
 Aduwati Sali (UPM, Malaysia)
 Jiehua Chen (Seimens, China)
 Tomoki Watanabe (TIT, Japan)
 Kazushi Nakano (UEC, Japan)
 Lin Liu (UNISA, Australia)
 Hoi-Shun Antony Lui (UQ, Australia)
 Konstanty Bialkowski (UQ, Australia)
 Kesorn Weaver (U of Southampton, UK)
 Dan Juncu (Cambridge Consultants, UK)
 Sarah Lee (Mango Solutions, UK)
 Woon Hau Chin (Toshiba Lab, UK)
 Kenedy Ailia Greyson (DIT, Tanzania)
 Xiaobing Hou (CCSU, USA)
 Shuju Wu (CCSU, USA)
 Chatree Sangprachathanarak (NetApp, USA)

Call for Papers:

The program of ECTI-CON 2014 will consist of regular technical sessions and special sessions. Topics of interest for submission include, but are not restricted to:

Area 1) Device, Circuits and Systems: Semiconductor Devices, Analog Circuits, Digital Circuits, Mixed Signal Circuits, Nonlinear Circuits and Systems, Sensing and Sensor Networks, Filters and Data Conversion Circuits, RF and Wireless Circuits, Photonic and Optoelectronic Circuits, Low Power Design and VLSI Physical Design, Biomedical Circuits, Assembly and Packaging Technologies, Test and Reliability, Advanced Technologies (i.e. MEMS and Nano-electronic Devices, Metamaterials), Agrionics, Embedded Systems;

Area 2) Computers: Computer Architecture, Computational Biology and Bioinformatics, Knowledge and Data Engineering, Learning Technologies, Multimedia Services and Technologies, Mobile Computing, Parallel/Distributed Computing and Grid Computing, Pattern Analysis and Machine Intelligence, Software Engineering, Visualization and Computer Graphics;

Area 3) Information Technology: IT BioMedical Engineering, Bioinformatics and Applications, Ontology, Business and Information Systems, Information Security and Forensics, Information Retrieval, Data Mining, Knowledge Management, Electronic Commerce, Health and Medical Informatics, Hybrid Information Technology

Area 4) Communication Systems: Communication Theory and Information Theory, Antenna and Propagation, Microwave Theory and Techniques, Modulation, Coding, Channel Analysis, Network Design, Network Protocols, Networks Management, Optical Communications, Wireless/Mobile Communications & Technologies, Green Wireless Networks, Green Radio;

Area 5) Controls: Control Theory and Applications, Adaptive and Learning Control Systems, Fuzzy and Neural Control, Mechatronics, Manufacturing Control Systems and Applications, Process Control Systems, Robotics and Automation;

Area 6) Electrical Power Systems: Power Engineering and Power Systems, Electromagnetic Compatibility, Energy Conversion, High Voltage Engineering and Insulation, Power Delivery, Illumination, Renewable/Alternative Energy, Energy Policy and Planning;

Area 7) Power Electronics: Power devices and components, Power Quality Control, Harmonic Analysis and Compensations, Switching Circuits and Power Converters, Motors and Drives, Smart Grid, Distribution Generation and Electrical Vehicles, Photovoltaic Materials and Solar Cells;

Area 8) Signal Processing: Signal Processing Theory, Digital Signal Processing Algorithms, Digital Filter Design & Implementation, Array Processing, Adaptive Signal Processing, Audio, Speech and Language Processing, Image and Video Processing, Medical Signal Processing, Medical Imaging;

Paper Submission: The review process of ECTI-CON2014 is "double-blind". Therefore, papers must be submitted without author's names and affiliations appearing in the manuscripts submitted for review. Prospective authors are invited to submit original full papers without author's names and affiliations, in English, of four to six (4-6) pages in standard IEEE two-column format only, reporting their original work and results, applications and/or implementation in one or more of the listed topics. Paper format (Doc/LaTeX) can be downloaded at <http://www.ecticon2014.org>.

Important dates

- Full paper submission due date: January 6, 2014
- Notification of acceptance: March 6, 2014
- Camera-ready paper submission: April 1, 2014
- Authors and Early-bird registration: April 1, 2014

Website: <http://www.ecticon2014.org>
<http://www.facebook.com/EctiCon2014>

Contact Address:
 School of Telecommunication Engineering
 Suranaree University of Technology
 Nakhon Ratchasima, Thailand 30000
 Tel: +66 4422 4392, Fax: +66 4422 4603



ECTI
 Association

IEEE
 THAILAND SECTION

Institute of Engineering
 Suranaree University of Technology



A Modified Particle Swarm Optimization with Dynamic Mutation Period

Chiabwoot Ratanavilisagul

Dept. of Computer Engineering
Faculty of Engineering, King Mongkut's Institute of
Technology Ladkrabang (KMITL), Bangkok, Thailand
chaibwoot@hotmail.com

Boontee Kruatrachue

Dept. of Computer Engineering
Faculty of Engineering, King Mongkut's Institute of
Technology Ladkrabang (KMITL), Bangkok, Thailand
boontee@yahoo.com

Abstract— the particle swarm optimization (PSO) is an algorithm that attempts to search for better solution in the solution space by attracting particles to converge toward a particle with the best fitness. PSO is typically troubled with the problems of trapping in local optimum and premature convergence. In order to overcome both problems, we propose an improved PSO algorithm that is applied mutation operator dynamically when particles are in local optimum. Moreover, the mutation period can be adjusted to solve the problem appropriately. The proposed technique is tested on benchmark functions and gives more satisfied search results in comparison with PSOs for the benchmark functions.

Keywords— *Swarm Intelligence; Particle Swarm Optimization; Cauchy Mutation; Mutation Operator.*

I. INTRODUCTION

Kennedy and Eberhart [1], [2] introduced Particle Swarm Optimization (PSO) in 1995. It is motivated by the behavior of flying bird and their communication mechanism in solving optimization problems. In comparison with several other population-based stochastic optimization methods, such as genetic algorithms (GAs) and evolutionary programming (EP), PSO performs better in solving a variety of optimization problems with rapid and stable convergence rate [3]-[5]. In addition, PSO is capable of solving the function optimization problems [6]-[8], neural network training [9], [10], as well as pattern classification and fuzzy system control [11], [12].

PSO has many advantages such as rapid convergence, simplicity, and little parameters to be adjusted. Its main disadvantage is having chances to converge on a local optimum, which leads to premature convergence. To overcome this disadvantage, many researchers [14]-[21] increased searching diversity in the population of PSO by adding the mutation operators in the process of PSO. The result showed that the proposed method can increase PSO optimization performance.

However, the mutation operators encounter parameters adjusting to suitably deal with any given problem since inappropriate parameters could contribute to poorer results than the standard PSO. The essential parameter of mutation operators is the mutation period.

Researchers proposed various ways to define the mutation period based on constant mutation rate [17], the best particle that is stagnant [17], and particle diversity [18].

This paper investigates adjusting mutation period of PSO to suitably deal with any given problem. Normally, the mutation should occur when swarm traps in local optimum. Hence, this paper proposes the mutation period defined by the unchanged number of the best consecutive particle which indicates the state of local optimum trapping. Moreover, mutation period is automatically adjusted according to the difference of the best solution found after mutation and the previous best solution.

A set of benchmark test functions is used to compare the standard PSO, the proposed PSO algorithm, PSO with mutation based on constant mutation rate (PSOMC) [17], PSO with mutation based on the best particle that is stagnant (PSOMS) [17], and PSO with mutation based on particle diversity (PSOMD) [18]. The results show that the proposed PSO algorithm obtains the best results in all test functions.

The rest of this paper is organized as follows. Section 2 explains basic PSO, the mutation operator, and elements of mutation operator. Section 3 explains PSO with dynamic mutation period (the proposed PSO algorithm). Section 4 explains the benchmark functions, the experiment setup and presents the experiment results. Section 5 concludes the paper with a brief summary.

II. RELATED WORK

A. Particle Swarm Optimization

In standard PSO, each member of the population is called a "particle" with its own position and velocity. Each individual particle performs searching in the search space according to its velocity, the best position found in the whole swarm (GBEST) and the individual's best position (PBEST). The standard PSO algorithm starts with randomizing particle positions and their respective velocities, and the evaluation of the position of each particle is achieved by using the objective function of the optimization problem. In a given iteration, each individual particle updates its position and velocity according to the expression below:

$$V_{id}' = \omega V_{id} + \eta_1 \text{rand}() (P_{id} - X_{id}) + \eta_2 \text{rand}() (P_{gd} - X_{id}) \quad (1)$$

$$X_{id}' = X_{id} + V_{id}' \quad (2)$$

Where X_{id} represents the current positions of i particle and d dimension, X_{id} represents the previous positions of i particle and d dimension, V_{id} represents the previous velocity of i particle and d dimension, V_{id}' represents the current velocity of i particle and d dimension, P_{id} represents PBEST of i particle and d dimension, and P_{gd} represents GBEST of d dimension. ω and η_1 and η_2 are acceleration constants, $0 \leq \omega < 1$ is an inertia weight and $\text{rand}()$ generates random number from interval [0,1]. A limit velocity is represented with V_{\min} , V_{\max} . If calculate velocity of a particle exceeds this value, it will replace value of V_{\min} , V_{\max} .

B. Mutation Operators

The mutation operator is the concept of genetic algorithm (GA). The principle of mutation operator is to randomly change positions of particles a little in some dimension within population. The lack of population diversity in PSO algorithms is a factor in their convergence on local optimum [13]. The mutation operator is added in process of PSO in order to increase population diversity. Therefore, it expands the search space and prevents convergence to local optimum [22]. It also increases the chance to escape from local optimum, so it improves the effectiveness of PSO [23].

C. Elements of Mutation Operators

The mutation applied to PSO has elements of mutation as follows:

1) The mutation equation is the equation designed for computing mutation to apply to some dimension of position or velocity in PSO. Many researchers proposed various mutation equations such as: N. Higashi and H. Iba. [20] implement a mutation equation from Gaussian distribution, so it is called Gaussian mutation. A. Stacey, M. Jancic, and I. Grundy. [21] implement a mutation equation from Cauchy distribution, so it is called Cauchy mutation. The results from different mutation equation affect the problem solving differently such as Cauchy mutation is suitable for multimodal problems while Power mutation is suitable for unimodal problems [16]. In order to solve multimodal problems efficiently, this paper selects Cauchy mutation.

2) The mutation point is position (particle positions, PBEST, GBEST) or velocity which can be mutated. Normally, the mutation is applied to particle positions.

3) The mutation amount is a number of dimensions of particles in population which is mutated in each mutation period. Many researchers proposed various mutation amounts such as: MO Lin. and ZHENG Hua. [18] used P_m to determine the mutation amount. Each dimension of particles in population will be mutated when random value on interval [0, 1] is less than P_m . P_m is taken as random value on interval [0.1, 0.3]. Andrews P S. [17] used mutation probability to determine the mutation amount. Each dimension of particles in population will be mutated when random value on interval [0, 1] is less

than mutation probability. The mutation probability is defined by using the following (3). Normally, the mutation amount is float point, so adjusting this parameter to suitably deal with any given problem is complicated.

$$M_p = \frac{M_r}{M \times D} \quad (3)$$

Where M represents the size of the population, D represents the dimension of the solution space, M_r represents the mutation probability, and M_r represents the mutation rate.

4) The mutation period is condition to decide which iteration is executed mutation or not. If the setting of this parameter causes over-mutation, this obstructs the convergence of particles and thus induces a random search, the result of which is poorer than that of standard PSO. On the other hand, under-mutation, provide too little mutation to escape from trapping in local optimum, thereby mutation is useless. Many researchers suggested setting of the mutation period in order to improve performance of the searching of PSO such as:

Andrews P S. [17] proposed PSO with mutation based on constant mutation rate (PSOMC). The concept of this algorithm can be summarized as follows: The mutation is applied in every iteration after all the particle positions have been updated. For each particle and for each dimension of a particle conditionally mutate the particle position in this dimension equal to mutation probability according to (3). The mutation probability depends on the number of particles and dimensions per particle.

The object of PSOMC is to enhance the population diversity by applying mutation in every iteration. Since mutation is applied in every iteration, this can be over-mutation. Some problems may not need to apply mutation or apply mutation in some iteration. Hence, the results of this algorithm may be poorer than those of standard PSO.

Andrews P S. [17] proposed PSO with mutation based on the best particle that is stagnant (PSOMS) [17]. The concept of this algorithm can be summarized as follows: Applying mutation is similar to PSOMC except the mutation is applied when the number of iteration that GBEST remains unchanged, is greater than some threshold value.

The object of PSOMS is to create particles jump out local minima when swarm traps in local optimum. This algorithm could adjust mutation period suitably for solving problems. The main problem of this algorithm is to identify threshold value suitable for a problem. If threshold value is inappropriately defined, the results of this algorithm may be poorer than those of standard PSO.

MO Lin. and ZHENG Hua. [18] proposed PSO with mutation based on particle diversity (PSOMD). The concept of this algorithm can be summarized as follows: The mutation is applied in each iteration after all the particle positions have been updated if the population average distance amongst points ($D(t)$) [24] is less than some threshold (D_{\min}). For each particle and for each dimension of particle, if a random number uniformly generated in the range [0:1] is less than P_m (P_m is random value on interval [0.1, 0.3]), then the particle position in this dimension is mutated.

The object of PSODM is to preserve the population diversity. This algorithm enhances the population diversity when it detects population diversity less than threshold value (D_{min}). The main problem of this algorithm is to identify threshold value suitable for a problem. If threshold value is inappropriately defined, the results of this algorithm may be poorer than those of standard PSO. The population average distance amongst points ($D(t)$) is described as following:

$$D(t) = \frac{1}{M \times L} \sum_{i=1}^M \sqrt{\sum_{d=1}^D (x'_{id} - p_{id})^2} \quad (4)$$

Where L represents the diagonal length of search space, M represents the size of the population, D represents the dimension of the solution space, x'_{id} represents the d dimension coordinate values of the i particle, and p_{id} represents the average value of the d dimension coordinate values of all particles.

III. PSO WITH DYNAMIC MUTATION PERIOD

As previously mentioned, the main problem of applying mutation operator technique is parameters adjusting. To avoid this problem, the proposed algorithm automatically adjusts its parameters.

The trapping in local optimum is possible for standard PSO, which leads to the stagnation of the searching in which the solution (GBEST) obtained at the point of trapping, is repeatedly produced irrespective of the length of searching time. Therefore, the unchanged number of consecutive GBEST can indicate state of swarm when swarm traps in local optimum.

Normally, the mutation operator should be used when swarm trap or trend to trap in local optimum in order to distribute particles to search in other areas. The easy techniques to indicate the trapping state of the swarm is to monitor the unchanged number of consecutive GBEST. Thereby, this paper proposed the mutation period considered from the unchanged number of consecutive GBEST.

If mutation period was set too small, particles will be reset before convergence. On the other hand, if mutation period was set too large, particles remain trapping in local optimum instead of search in other areas.

Therefore, this paper proposes the novel technique which can decrease problem of adjusting mutation period. The proposed technique can adjust the mutation period to solve any optimization problems appropriately.

The concept of proposed technique is explained as follows: before mutation, if the best solution in current mutation round is worse than previous round GBEST, particles could not converge to the GBEST. Hence, the mutation period should be increased to enhance convergence time of particles to the GBEST in next mutation round. On the other hand, if the best solution in current mutation round equals GBEST, particles can converge to the GBEST. In this case, the mutation period should be decreased to reduce the searching time in local optimum. The proposed technique is called PSO with dynamic

mutation period (PSODMP). Pseudo code of PSODMP is shown below:

```

Initial position and velocity of each particle
Initial PBEST, GBEST, RBEST
While termination condition ≠ true do
  For each particle
    Evaluate particle fitness
    If fitness of each particle is better than that of PBEST
      Update PBEST (Pi=Xi)
    End If
    If fitness of each particle is better than that of GBEST
      Update GBEST (Pg=Xi)
    End If
    If fitness of each particle is better than RBEST
      Update RBEST (RBEST=Xi)
      TRBEST = -1
    End If
    Update particle position according to formula (1) and (2)
  End For
  TRBEST ++
  If TRBEST >= MP
    If fitness of RBEST is worse than fitness of GBEST
      MP++
    Else
      MP--
    End If
    If MAXMP < MP
      MP = MAXMP
    End If
    If MINMP > MP
      MP = MINMP
    End If
    Reset RBEST
    For each particle
      If percent mutation < rand() then
        Apply mutation according to formula (5)
      End If
    End For
    TRBEST = 0
  End If
End while

```

Equation (5) is mutation equations used in PSODMP in which the positive operator is selected if the random number generated uniformly in the range [0, 1] is less than 0.5 and the negative operator otherwise

$$\text{mutation}(X_{id}) = X_{id} \pm \text{Cauchy}(\delta) \quad (5)$$

Where δ is a scale parameter that determines the shape of the distribution. Cauchy (δ) denotes Cauchy random numbers with the scale parameter of 1. The upper and lower limit positions are X_{\max} , X_{\min} . MP is the mutation period. RBEST is the best solution in each mutation round. TRBEST is the times of RBEST consecutive unchanged. MINMP is the minimum value of the mutation period. MAXMP is the maximum value of the mutation period.

IV. EXPERIMENTS AND RESULTS

A. Benchmark Functions

The proposed algorithm is tested on ten well-known benchmark functions, listed in Table 1. These functions consist of seven multimodal functions from function one to seven. The remaining three functions are unimodal functions. The results of PSODMP on the benchmark functions are compared with PSO, PSOMC, PSOMS, and PSOMD.

TABLE I. DETAILS OF BENCHMARK TEST FUNCTIONS.

Problem no.	Function name	Expression
1	Ackley	$f(x) = 20 \exp\left(0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n f_i}\right) \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + n$
2	Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(x_i / \sqrt{i}\right) + 1$
3	Rastrigin	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$
4	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
5	Schweffel	$f(x) = 418.9829 \times n + \sum_{i=1}^n (x_i \times \sin(\sqrt{ x_i }))$
6	Step	$f(x) = \sum_{i=1}^n (x_i + 0.5)^2$
7	Conic Mixture	$f(x) = -0.1 \times \sum_{i=1}^n \cos(5\pi x_i) + \sum_{i=1}^n x_i^2 + 0.1n$
8	Sphere	$f(x) = \sum_{i=1}^n x_i^2$
9	Parallell Ellipsoid	$f(x) = \sum_{i=1}^n (i \times x_i^2)$
10	Rotated Ellipsoid	$f(x) = \sum_{i=1}^n (i \times x_i^2)$

TABLE I. (CONTINUED).

Problem no.	Search space [X_{\min} , X_{\max}]	Objective function value	Dim.	V_{\max} , V_{\min}
1	$x \in [-32.768, 32.768]^n$	0	30	-32.768, 32.768
2	$x \in [-300, 300]^n$	0	30	-300, 300
3	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
4	$x \in [-2.048, 2.048]^n$	0	30	-2.048, 2.048
5	$x \in [-500, 500]^n$	0	30	-500, 500
6	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
7	$x \in [-1, 1]^n$	0	30	-1, 1
8	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
9	$x \in [-5.12, 5.12]^n$	0	30	-5.12, 5.12
10	$x \in [-65.536, 65.536]^n$	0	30	-65.536, 65.536

B. Parameters Setting

Parameters are as follows for all experiments: η_1 and η_2 are both set to be 1.496180 and $\omega = 0.729844$, as suggested by F. van den Bergh [25], the number of population is 80. The

number of experiments of each function is 100 runs. The maximum iteration is 10000. To guarantee fairness of performance measurement, the mutation equation (5) is applied to all methods.

The non-PSO parameters are as follows: For PSODMP algorithm, MINMP = 2, MAXMP = 100, MP = 50, the mutation probability = 0.01. The compared algorithm parameters are set according to suggested by the original papers. For PSOMC algorithm, the mutation rate = 0.5. For PSOMS algorithm, the mutation rate = 0.5, threshold value = 5. For PSOMD algorithm, $D_{\min} = 0.001$.

This research is conducted by a personal computer of Intel Core i5 2450 with a 2.5-GHz CPU and 8 GB RAM, and Visual C++ as the programming language.

In Table 2, the mutation period of PSODMP is not adjusted dynamically, but it is fixed in order to compare to results from mutation period adjusting of PSODMP. To guarantee fairness of performance measurement, numbers of evaluation of all the algorithms are set equation. The number of evaluation of each run in experiments is 800000 evaluations.

TABLE II. THE RESULTS OF PSODMP BY FIXED MP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO	PSODMP	PSODMP	PSODMP	PSODMP	PSODMP
MP	-	2	5	10	50	100
Problem no.	MBF	MBF	MBF	MBF	MBF	MBF
1	0.330091	7.28E-15	7.57E-15	7.92E-15	8.67E-15	8.74E-15
2	0.0119302	0.00167542	0.00640026	0.00838687	0.0116821	0.0117401
3	7.324E-7	0.0320216	1.71E-15	7.39E-15	5.57E-14	7.16E-14
4	3.32E-10	0.011615	2.70E-19	1.34E-29	7.64E-31	3.45E-31
5	2008.25	37.1888	59.3283	170.483	693.962	940.344
6	6.00E-02	0	0	0	0	0
7	0.378328	0	0	0	3.55E-17	5.33E-17
8	1.46E-251	9.32E-219	1.20E-245	1.04E-250	8.99E-251	7.40E-252
9	1.57E-249	3.74E-217	8.84E-248	1.64E-249	2.24E-249	2.45E-249
10	2.70E-255	6.08E-223	1.67E-253	2.02E-255	1.34E-257	9.30E-256

The measures of algorithm performance in the experiments are as follows: The mean best fitness value (MBF) is the mean of best fitness in the final iteration from all running (100 runs). MBF indicates the solution searching efficiency of an algorithm. All experiments benchmark functions have zero results as a minimum point. In Table 1, an entry less than 10^{-323} is given the value of zero. The closer the MBF to the zero point of a method, the better the method. The mean mutation period (MMP) is the mean of mutation period in every iteration from all running. MMP is the results of the calculation of PSODMP algorithm. SD is the standard deviation. SD indicates the solution searching reliability of an algorithm. MD(t) is the mean of the population average distance amongst points in the final iteration from all running. MD(t) indicates the diversity of algorithms. Student's T-Test (ST) is a statistical method that is used to see if two sets of data differ significantly.

From the experimental results of Table 2 show some functions, if the setting mutation period is 2 or 5, swarm will be over-mutation. Its searching solutions are poorer than results of the standard PSO, such as Rosenbrock function, Sphere function, Parallel Ellipsoid function, and Rotated Ellipsoid function. This is due to particles converge hardly. In the case of mutation period is equal or more than 50 in unimodal functions, overall its searching solutions are similar to results of the standard PSO since the mutation operator is not executed. For multimodal problems, if the setting mutation period is more than 50, swarm will be under-mutation. Its overall results are poorer than results of the mutation period setting ranged from 5 to 10, such as Ackley function, Griewank function, Rastrigrin function, Schwefel function, and Cosine function. Since swarm wastes time searching in local optimum rather than starting new search in other areas.

TABLE III. COMPARATIVE RESULTS OF PSO, PSOMC, PSOMS, PSOMD AND PSODMP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO		PSOMC		PSOMS		PSOMD		PSODMP	
	MBF	MBF	MBF	MBF	MBF	MBF	MBF	MBF	MBF	MBF
1	0.3300E1	8.99E-15	8.99E-15	7.06E-09	7.53E-15	2.304E2				
2	0.0193E2	0.0123E09	0.00955E5	0.0022653	0.001527E7	2.59979				
3	7.524E7	1.66E-14	4.43E-14	9.58144	5.41E-15	6.569E5				
4	5.32E-39	2.96E-31	4.44E-31	0.000106E23	1.48E-31	51.1689				
5	200E-25	1424.73	1423.54	1303.92	7.46E-01	2.4443				
6	6.89E-02	0	0	0.01	0	88.6433				
7	0.3783E8	0	2.66E-17	1.97E-13	0	10.8324				
8	1.46E-251	1.29E-246	2.51E-249	1.03E-15	4.63E-251	50				
9	1.57E-249	3.51E-246	3.88E-248	9.85E-15	2.34E-249	50				
10	2.70E-255	5.75E-247	1.46E-254	5.14E-231	2.52E-257	50				

For multimodal problems in Table 2, PSODMP gets good results because of the mutation period setting ranged from 2 to 50. The PSODMP algorithm can adjust mutation period in this range and it can obtain the good solutions as shown in Table 3. Therefore, the PSODMP can dynamically adjust to suitably deal with the problem.

From the experimental results of Table 3 show that PSODMP outperforms PSO, PSOMC, PSOMS, and PSOMD in multimodal function. PSODMP produces a better quality solution because of its lowest MBF in multimodal function.

For multimodal function in Table 3, some MBF results of PSOMD are poorer than the standard PSO such as Rosenbrock function. If mutation period adjusting to deal with problems appropriately. The solutions of PSOMD can obtain better than that of the standard PSO. However, mutation period adjusting of PSOMD to deal with any given problem appropriately is complicated.

For unimodal function in Table 3, PSO and PSODMP have similar overall results. The MBF results of PSO and PSODMP are higher than that of PSOMC, PSOMS, and PSOMD because these algorithms could execute the mutation operator which

will hinder the convergence of particles in unimodal function. In case of unimodal function shows PSOMC, PSOMS, and PSOMD cannot adjust mutation period to deal with any given problem appropriately. Hence, the results of these algorithms could be poorer than standard PSO.

TABLE IV. COMPARATIVE RESULTS OF SD OF PSO, PSOMC, PSOMS, PSOMD AND PSODMP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO		PSOMC		PSOMS		PSOMD		PSODMP	
	SD	SD	SD	SD	SD	SD	SD	SD	SD	SD
1	0.5836E5	3.15E-15	3.19E-15	3.83E-09	2.42E-15					
2	0.0128E51	0.0163161	0.0105423	0.0056868	0.0053869					
3	19.5677	5.53E-14	4.53E-14	14.799E	1.55E-14					
4	1.58E-29	4.61E-30	5.48E-30	4.34E-05	3.50E-30					
5	569.192	341.53E	365.627	32E-79	81.61E					
6	0.698212	0	0	0.908E25	0					
7	0.20742	0	1.59E-16	1.25E-13	0					
8	0	0	0	8.72E-16	0					
9	0	0	0	1.63E-14	0					
10	0	0	0	0	0					

TABLE V. COMPARATIVE RESULTS OF MD(T) OF PSO, PSOMC, PSOMS, PSOMD AND PSODMP ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Techniques	PSO		PSOMC		PSOMS		PSOMD		PSODMP	
	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)	MD(T)
1	2.9E-24	1.618E-14	1.6956E-16	0.2036E3	5.83E-06					
2	0.0038E719	0.0310E8	0.0538E4	0.1313E7	0.3308E6					
3	2.71834E-17	1.93217	0.294151	0.02777E3	0.679165					
4	3.39155E-31	0.0124E34	0.005104	0.21838	0.009747					
5	2.3827E-14	2.33763	2.47937	35.6269	591.101					
6	0.023609E	0.126497	0.088937	0.0324901	0.076241					
7	3.0790E-17	1.97259E-09	7.58E-09	0.00408327	4.1E-09					
8	1.4003E-251	6.95328E-17	5.79E-251	0.009220E5	2.6E-250					
9	5.6331E-251	4.6627E-137	2.34E-248	0.008914E	2.1E-250					
10	1.3000E-08	0.279225	0.000365	0.42919	3.8E-05					

TABLE VI. COMPARATIVE RESULTS OF ST BETWEEN PSODMP AND THE OTHER ALGORITHMS (PSO, PSOMC, PSOMS, PSOMD) ON BENCHMARK TEST FUNCTIONS AVERAGED OVER 100 RUNS.

Compared algorithms	PSO	PSOMC	PSOMS	PSOMD
	ST	ST	ST	ST
PSODMP	0.162501	0.17171	0.17171	0.169745

From the experimental results of Table 4 show that the reliability of PSODMP is better than that of PSO, PSOMC, PSOMS, and PSOMD because of its lowest SD in all test functions. Hence, PSODMP outperforms PSO, PSOMC, PSOMS, and PSOMD in case of reliability and quality solution.

For multimodal problems in Table 5, the population diversity of PSOMC, PSOMS, PSOMD and PSODMP are more than that of PSO because MD(t) of PSOMC, PSOMS, PSOMD and PSODMP in all multimodal functions are more than that of PSO. Hence, the mutation operator can improve the population diversity of PSO.

From the experimental results of Table 6 show that the statistically significant result of PSODMP is different from that of PSO, PSOMC, PSOMS, and PSOMD. Therefore, the ST results of PSO, PSOMC, PSOMS, and PSOMD have a little value when these algorithms are compared with PSODMP.

V. CONCLUSION

The mutation operator is applied with PSO to solve problems of trapping in local optimum and premature convergence. However, applying the mutation operator with general problems is complicated which is the cause of complicatedly adjusting the mutation period in order that the mutation operator can enhance the solution searching performance and can be easily applied to general problems. This paper proposes the mutation period that is automatically adjusted according to the difference of the best solution found after mutation and the previous best solution. The proposed technique is called PSODMP. On benchmark functions, the proposed PSODMP, PSO, PSOMC, PSOMS, and PSOMD are tested and the results are compared. From the experimental results show that PSODMP can adjust the mutation period to solve the problem appropriately and PSODMP outperforms PSO, PSOMC, PSOMS, and PSOMD.

REFERENCES

- [1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
- [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*, 1995, pp. 39-43.
- [3] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, M. N. Vrahatis, "Objective function stretching to alleviate convergence to local minima," *Nonlinear Analysis TMA* 47, 2003, pp. 3419-3424.
- [4] R. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *The 7th Annual Conference on Evolutionary Programming*, San Diego, 1998.
- [5] J. Kennedy, R.C. Eberhart, "Swarm Intelligence, Morgan Kaufmann," San Mateo, 2001.
- [6] V.S. Pappala, I. Erlich, K. Rohrig, J.A. Dobschinski, "A stochastic modal for the optimal operation of a wind-thermal power system," *IEEE Trans. Power Syst.*, May 2009, pp. 940-950.
- [7] P. Acharjee, S.K. Goswami, "A decoupled power flow algorithm using particle swarm optimization technique," *Energy Convers. Manage.*, 2009, pp. 2351-2360.
- [8] S. Dutta, S.P. Singh, "Optimal rescheduling of generators for congestion management based on particle swarm optimization," *IEEE Trans. Power Syst.*, 2008.
- [9] S. Kiranyaz, T. Ince, A. Yildirim, M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," *Neural Networks*, 2009, pp. 1448-1462.
- [10] H.L. Wei, S.A. Billings, Y.F. Zhao, L.Z. Guo, "Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification," *IEEE Trans. Neural Network*, 2009, pp. 181-185.
- [11] C.J. Lin, C.H. Chen, C.T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *IEEE Trans. Syst. Man Cybernetics Part C*, 2009, pp. 55-68.
- [12] L. Zhao, F. Qian, Y. Yang, Y. Zeng, H. Su, "Automatically extracting TS fuzzy models using cooperative random learning particle swarm optimization," *Appl. Soft Comput.*, 2010, pp. 938-944.
- [13] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, 2004, pp. 240-255.
- [14] R. A. Krohling, "Gaussian particle swarm with jumps," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Edinburgh, UK, 2005, pp. 1226-1231.
- [15] R. A. Krohling, L. dos Santos Coelho, "PSO-E: Particle Swarm with Exponential Distribution," *Proceedings of the IEEE Congress on Evolutionary Computation*, July 2006, pp. 1428-1433.
- [16] Muhammad Inuran, Hajira Jabeen, Mubashir Ahmad, Qamar Abbas, Waqas Bangyal, Qamar Abbas, "Opposition based PSO and Mutation Operators," *2010 2nd International Conference on Education Technology and Computer (ICETC)*, pp. 506-508.
- [17] Andrews P. S., "An investigation into mutation operators for particle swarm optimization," *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, 2006, pp. 1044-1051.
- [18] MO Lin, ZHENG Hua., "Improved PSO Algorithm with Adaptive Inertia Weight and Mutation," *2009 World Congress on Computer Science and Information Engineering*, 2009, pp. 622-625.
- [19] Chiraboot Ratanavilisagul, Boontee Kruatrachue, "Mutation Period Calculation for Particle Swarm Optimization," *1st International Symposium on Technology for Sustainability*, 2011, pp. 213-216.
- [20] N. Higashi and H. Iba., "Particle swarm optimization with Gaussian mutation," *Proc. of the 2003 IEEE Swarm Intelligence Symposium*, 2003, pp. 72-79.
- [21] A. Stacey, M. Jancic, and I. Grundy., "Particle swarm optimization with mutation," *Proc. of the 2003 IEEE Congr. On Evol. Comput.*, 2003, pp. 1425-1430.
- [22] D. Dumitrescu, B. Lazerini, L.C. Jain, A. Dumitrescu, "Evolutionary Computation," CRC Press, 2000.
- [23] Hanyong Choi, Shunichi Ohmori, Kazuho Yoshimoto, Hiroaki Ohtake, "Improvement of Particle Swarm Optimization Application of the Mutation Concept for the Escape from Local Minima," *8th International Conference on Supply Chain Management and Information Systems*, 2010, pp. 1-5.
- [24] KRINK T, VESTERSTROM J S, RIGET J., "Particle swarm optimization with spatial particle extension," *Proceedings of IEEE Congress on Evolutionary Computation*, Honolulu, Hawaii USA, 2002.
- [25] F. van den Bergh., "An Analysis of Particle Swarm Optimizers," PhD thesis, Department of Computer Science, University of Pretoria, South Africa, 2002.

ผลงานวิจัยที่ได้รับการตีพิมพ์ (Short Paper)

3. Chiabwoot Ratanavilisagul, Boontee Kruatrachue, “A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period”, International Conference on Computing and Information Technology (IC2IT), Volume 265, 8-9 May 2014, pp. 33-42



The 10th International Conference on Computing and Information Technology (IC²IT2014) May 8-9, 2014

Angsana Laguna, Phuket, Thailand

<http://www.ic2it.org>

Computing and Information Technology Drives Societies: Theory and Applications

General Chair

Phayung Meesad, KMUTNB, Thailand

Program Committee Chair

Herwig Unger, FernUni, Germany

Program Committee Members

M Aiello, Groningen, the Netherlands
 T Bernard, Université de Reims, France
 W Bothsawan, KU, TH
 S Boonkrong, KMUTNB, TH
 T Böhme, TU Ilmenau, Germany
 A Bui, Université Paris 8, France
 M Caspar, Chemnitz, Germany
 D Delen, OSU, USA
 T Eggendorfer, Hamburg, Germany
 G Eichler, Telecom, Germany
 P Guenther, H3 Esslingen, Germany
 R Gunzaj, Uni Maribor, Slovenia
 H C Ha, HNUE, Vietnam
 M Hagan, OSU, USA
 W Harot, Chemnitz, Germany
 C Haruechayasak, NECTEC, TH
 K Hengrathrom, NFRU, TH
 J Hacprzyk, Polish Acad. of Sci., Poland
 P Kropf, Neuchâtel, Switzerland
 F Yamakya, Hagenfurt, Austria
 B Lenz, Lenz AG, Switzerland
 U Lechner, UnBw, Germany
 W Limpakorn, NECTEC, TH
 N T Loc, HNUE, Vietnam
 J Lu, UTS, Australia
 A Mikler, UNF, USA
 A Mingkhwan, KMUTNB, TH
 S Nitsuwat, KMUTNB, TH
 S Nuamneesi, SSRU, TH
 P Prathombur, NECTEC, TH
 A Preechayasonboon, TOT, TH
 G Quirchmayr, UNIVIE, Austria
 C Ramirez, USL, Mexico
 T Sikkhacha, TOT, Thailand
 W Sruai, UBU, TH
 W Tang, CityU, Hongkong
 T Till, Telecom, Germany
 D H Tran, HNUE, Vietnam
 H T sai, NTU, Taiwan
 D Tutsch, Wuppertal, Germany
 N Viskipongpun, KMUTNB, TH
 K Vorakraisawan, FMITL, TH
 W Warakraisawan, BU, TH
 M Weiser, OSU, USA

Call for Papers

The huge amount of raw data generated by government, industry and business, commonly known as **Big Data**, require more and more efficient tools to turn them into useful information and knowledge. The 10th International Conference on Computing and Information Technology provides an exchange of the state of the art and future developments in the two key areas of this process: Computer Networking and Data Mining. Behind the background of the foundation of ASEAN, it becomes clear that efficient languages, business principles and communication methods need to be adapted, unified and especially optimized to gain a maximum benefit to the users and customers of future IT systems.

The organizing partners are King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand; Fern University in Hagen (FernUni), Germany; Chemnitz University of Technology (CUT), Germany; Edith Cowan University (ECU), Australia; Oklahoma State University (OSU), USA; Hanoi National University of Education (HNUE), Vietnam; Mahasarakham University (MSU), Thailand; Ubon Ratchathani University (UBU), Thailand; Kanchanaburi Rajabhat University (KRU), Thailand; Nakhon Pathom Rajabhat University (NPRU), Thailand; Mahasarakham Rajabhat University (RMU), Thailand; Rajamangala University of Technology Lanna (RMUTL), Thailand; Rajamangala University of Technology Krungthep (RMUTK), Thailand; Rajamangala University of Technology Thanyaburi (RMUTT), Thailand; Prince of Songkla University, Phuket Campus, Thailand; National Institute of Development Administration, Thailand and Council of IT Deans of Thailand (CITT).

Scope

Conference topics include (but not limited to):

- **Data Mining and Machine Learning:** Artificial Neural Network, Fuzzy Systems, Hybrid Systems, Evolutionary Computation, Knowledge Discovery, Knowledge Transfer, Knowledge Management, Decision Support, Recommender Systems, Text Mining, Web Mining, Human-Computer Interface and Image Processing.
- **Data Network and Communication:** Computer Network, Security & Forensic, Wireless & Sensor Network, Telecommunication, Mobile Ad-Hoc Network, Cloud & Grid Computing, Decentralized computing, P2P networks, P2P protocols, and Semantic P2P networks.

Important Dates

Paper Submission Deadline for Review: **December 15, 2013**

Decision Notification: **December 23, 2013**

Camera Ready Version: **January 13, 2014**

Advanced Registration: **January 20, 2014**

Paper Submission



Papers must be written in English and should describe original work in details. Each paper must follow the format, according to the instructions on the conference web site at <http://www.ic2it.org> or Springer web site at <http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0>.

Each paper must be accompanied by an abstract summarizing the contribution it makes to the field. Maximum paper length for regular papers is 10 pages (A4). The manuscript can be submitted at <http://www.easychair.org/conferences/?conf=ic2it2014>. Each paper will be reviewed by at least three reviewers. Submission of a paper constitutes a commitment that, if accepted, at least one author must attend and participate in the conference. Accepted papers will be published in the conference proceedings by ISI Proceedings, DBLP, Ulrich's, EI-Compendex, SCOPUS, Zentralblatt Math, MetaPress, Springerlink.

Contact Information

Assoc.Prof.Dr. Phayung Meesad, IC²IT 2014 General Chair

Prof.Dr. Herwig Unger, Fern University in Hagen, IC²IT2014 Technical Program Chair

Faculty of Information Technology

King Mongkut's University of Technology North Bangkok

1518 Pracharat 1 Rd., Wongsawang, Bangsue, Bangkok 10800, Thailand

Email: pym@kmutnb.ac.th; Tel: +662 555 2000 Ext.2711, 2719, 2726; Fax: +662 555 2734

A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period

Chiabwoot Ratanavilisagul and Boontee Kruatrachue

Department of Computer Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
chaibwoot@hotmail.com, boontee@yahoo.com

Abstract. The particle swarm optimization (PSO) is an algorithm that attempts to search for better solution in the solution space by attracting particles to converge toward a particle with the best fitness. PSO is typically troubled with the problems of trapping in local optimum and premature convergence. In order to overcome both problems, we propose an improved PSO algorithm that can re-initialize particles dynamically when swarm traps in local optimum. Moreover, the particle re-initialization period can be adjusted to solve the problem appropriately. The proposed technique is tested on benchmark functions and gives more satisfied search results in comparison with PSOs for the benchmark functions.

Keywords: Particle Swarm Optimization, Particles Re-initialization, Mutation operator, Multi-start Particles.

1 Introduction

Kennedy and Eberhart [1, 2] introduced Particle Swarm Optimization (PSO) in 1995. It is motivated from the behavior of flying bird and their communication mechanism in solving optimization problems. In comparison with several other population-based Stochastic optimization methods, such as genetic algorithms (GAs) and evolutionary programming (EP), PSO performs better in solving a variety of optimization problems with fast and stable convergence rate [3-5]. In addition, PSO is capable of solving the function optimization problems [6-8], artificial neural network training [9, 10], as well as pattern classification and fuzzy system control [11, 12].

PSO has many advantages such as rapid convergence, simplicity, and little parameters to be adjusted. Its main disadvantage is trapping in local optimum and premature convergence. To overcome both problems, many researchers [13-19] increased searching diversity in the population of PSO by adding the particle re-initialization or the mutation operators in the process of PSO. The result showed that both methods can increase PSO optimization performance.

However, the particle re-initialization encounter parameters adjusting to suitably deal with any given problem. As inappropriate parameters could contribute to poorer

results vis-à-vis the standard PSO. The essential parameter of particle re-initialization is the particles re-initialization period (re-initialization period) which is the amount of time to be certain that the swarm trapped in local optimum and the reinitialized particles with PSO should be performed.

Researchers proposed various ways to define re-initialization period using the number of iteration with the best position unchanged [14], and the number of iteration with the consecutive neighborhood best position unchanged [15].

This paper investigates adjusting re-initialization period to suitably deal with any given problem applied with PSO. Normally, the reinitialized particles (resetting) should occur when swarm traps in local optima. Hence, this paper proposes the re-initialization period defined by the unchanged number of consecutive best particle which indicates the state of local optima trapping. Moreover, re-initialization period is automatically adjusted according to the difference of the best solution found after resetting and the previous best solution. A set of benchmark test functions is used to compare the standard PSO, the proposed PSO algorithm, PSO algorithm's [14], and PSO algorithm's [15]. The results show that the proposed PSO algorithm obtains the best results in all test functions.

The rest of this paper is organized as follows. Section 2 explains basic PSO and PSO with particles re-initialization. Section 3 explains PSO with dynamic particles re-initialization period. Section 4 explains the benchmark functions, the experiment setup and presents the experiment results. Section 5 concludes the paper with a brief summary.

2 Related Work

2.1 Particle Swarm Optimization

In standard PSO, each member of the population is called a "particle" with its own position and velocity. Each individual particle performs searching in the search space according to its velocity, the best position found in the whole swarm (GBEST) and the individual's best position (PBEST). The standard PSO algorithm starts with randomizing particle positions and their respective velocities, and the evaluation of the position of each particle is achieved using the objective function of the optimization problem. In a given iteration, each individual particle updates its position and velocity according to the expression below:

$$V_{id}' = \omega V_{id} + \eta_1 \text{rand}() (P_{id} - X_{id}) + \eta_2 \text{rand}() (X_{pid} - X_{id}) \quad (1)$$

$$X_{id}' = X_{id} + V_{id}' \quad (2)$$

Where X_{id}' denotes the current positions of i particle and d dimension, X_{id} the previous positions of i particle and d dimension, V_{id} the previous velocity of i

particle and d dimension, V_{id}^t the current velocity of i particle and d dimension, P_{id}^t PBEST of i particle and d dimension, and p_{gd} GBEST of d dimension. $0 \leq \omega < 1$ is an inertia weight, η_1 and η_2 are acceleration constants, and $\text{rand}()$ generates random number from interval $[0,1]$. A limit velocity calls V_{\max} . If calculate velocity of a particle exceeds this value, it will replace value of V_{\max} .

2.2 PSO with Particles Re-initialization

The particle re-initialization techniques prevent premature convergence and trapping in local optimum. They create a variation of a population. Many researchers proposed various techniques such as:

Den Bergh [13] proposed Multi-start PSO (MPSO). The concept of this algorithm can be summarized as follows: after some iteration with the maximum swarm radius less than some threshold value, all particles of swarm are initialized in order to increase the diversity of swarm and expand the searching space. This algorithm can jump out of the local minimum and converge to the new area. However, this technique destroys the current structure of the swarm with total re-initialization of particles which results in decreasing the convergence speed and the search accuracy [14, 15].

In fact, the swarm structure is not totally destroyed because GBEST is not reset. As a result, the re-initialized swarm is position very far from the GBEST and will rapidly converge to the GBEST. If swarm can locate a better position than the previous GBEST, the better position is then occupied as a new GBEST. However, MPSO has difficulty in adjusting suitably re-initialization period. Applying MPSO with general problems is complicated. In addition, the results from non-reset PBEST contribute to non-distributed searching with rapid convergence, so MPSO is rarely improve the searching performance of PSO.

Ning Li et al. [14] proposed PSO with mutation by considering from the Euclidean distance and GBEST (PSOMEG). The concept of this algorithm can conclude as follows: The velocity and position of particles are initialized depend on mutation probability. The initialization occur when GBEST unchanged more than some the threshold value (MaxStep) and the maximum Euclidean distance between all the particles is less than the threshold value (BorderRadius). This technique improves the diversity of swarm without decreasing the convergent speed and the search accuracy. However, PSOMEG has three additional parameters from standard PSO. MaxStep must be in some range, otherwise the results is poor. BorderRadius is very difficult to set with any given problem. Mutation probability can be set to over-mutation which results in similar searching to MPSO. On the other hand, set to under-mutated results in remain trapping. The difficulty of parameters setting reduces the practicality of PSOMEG with general problems.

Xuedan Liu et al. [15] proposed PSO with mutation by considering from the neighborhood's best position (PSOMN). In this paper, the velocity and position of particles is initialized when the unchanged numbers of the consecutive neighborhood best position are more than some threshold value (MaxStep). PSOMN increase the

global search capability and avoid premature convergence. However, selection of MaxStep is recommended within some range to obtain good results.

3 PSO with Dynamic Particles Re-initialization Period

As previously mentioned, the main problem of applying particles re-initialization technique is parameters adjusting. To avoid this problem, the proposed algorithm automatically adjusts its parameters. In addition, the number of parameters is kept minimal.

The trapping in local optimum is possible for standard PSO, which leads to stagnation of the searching in which the solution (GBEST) obtained at the point of trapping, is repeatedly produced irrespective of the length of searching time. Therefore, the unchanged number of consecutive GBEST can indicate state of swarm when swarm traps in local optimum.

Normally, the particles re-initialization (resetting) should be used when swarm trap or trend to trap in local optimum in order to distribute particles to search in other area. The easy techniques to indicate the trapping state of the swarm is to monitor the unchanged number of consecutive GBEST. Thereby, this paper proposed the particles re-initialization period (re-initialization period) is considered from the unchanged number of consecutive GBEST.

If re-initialization period was set too small, particles will be reset before convergence. On the other hand, if re-initialization period was set too large, particles remain trapping in local optimum instead of search in other areas.

Therefore, this paper proposes the novel technique which can decrease problem of adjusting re-initialization period. The proposed technique can adjust the re-initialization period to solve any optimization problems appropriately. The concept of proposed technique is explained as follows: Before reset, if the best solution in current resetting round is worse than previous round GBEST, particles could not converge to the GBEST. Hence, the re-initialization period should be increased to enhance convergence time of particles to the GBEST in next resetting round. On the other hand, if the best solution in current resetting round equals GBEST, particles can converge to the GBEST. In this case, the re-initialization period should be decreased to reduce the searching time in local optimum. In addition, the proposed technique choose to initialize all particles to improve chance in obtain better solution than initializing some particles. Because the more the number of particles reset, the better chance in obtaining the new GBEST while particles return to GBEST. Furthermore, PBEST of all particles should be reset to enhance distribution of particles. The proposed technique is called PSO with dynamic particles re-initialization period (PSODR). Pseudo code of PSODR is shown below:

A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period 37

```

Initial position and velocity of each particle
Initial PBEST GBEST RBEST
While termination condition ≠ true do
  For each particle
    Evaluate particle fitness
    If fitness of each particle is better than that of PBEST
      Update PBEST ( $P_i = X_i$ )
    End If
    If fitness of each particle is better than that of GBEST
      Update GBEST ( $P_g = X_i$ )
    End If
    If fitness of each particle is better than RBEST
      Update RBEST (RBEST =  $X_i$ )
      Set the times of RBEST consecutive unchanged to 0
    End If
    Update particle position according to formula (1) and (2)
  End For
  Times of RBEST consecutive unchanged ++;
  If the times of RBEST consecutive unchanged >= PRP
    If fitness of RBEST is worse than fitness of GBEST
      PRP = PRP + NPRP
    Else
      PRP = PRP - NPRP
    End If
    If PRP is less than the MINPRP
      PRP = MINPRP
    Else If PRP is more than the MAXPRP
      PRP = MAXPRP
    End If
    Reset RBEST
    Reinitialize PBEST, velocity, and position of all particles
    Set the times of GBEST consecutive unchanged to 0
  End If
End while

```

PRP is the particle re-initialization period (re-initialization period). RBEST is the best solution in each resetting round. NPRP is adjusting round number of the particle re-initialization period (this is the dynamic adjustment of PRP). MINPRP is the minimum value of the particle re-initialization period. MAXPRP is the maximum value of the particle re-initialization period.

4 Experiments and Results

4.1 Benchmark Functions

The proposed algorithm is tested on nine well-known benchmark functions, listed in Table 1. These functions consists of six multimodal functions from function one to six. The remaining three functions are unimodal functions. The results of PSODR on the benchmark functions are compared with PSO [1, 2], PSOMEG [14] and PSOMN [15]. To guarantee fairness, the inertia weigh of PSOMN algorithm cannot be adapted as written in original paper. The inertia weigh of PSOMN algorithm is fixed as other algorithms.

4.2 Parameters Setting

Parameters are as follows for all experiments: acceleration constants of η_1 and η_2 are both set to be 1.496180 and inertia weight $\omega = 0.729844$, as suggested by den Bergh [13], the number of population is 20. The number of experiments of each function is 100 runs. The maximum iteration is 5000 independently of number of resetting. Hence, the algorithm that locates local optimum faster has more number of trial runs.

The non-PSO parameters are as follows: For PSODR algorithm, MINPRP = 10, MAXPRP = 1000, NPRP = 50, PRP = 200. The compared algorithm parameters are set according to suggested by the original papers. For PSOMEG algorithm, the mutation probability = 0.04, MaxStep = 10, the divided number in calculation of BorderRadius is 1000. For PSOMN algorithm, MaxStep = 30, the neighborhood length = 2.

Table 1. Details of Benchmark Test Functions.

Function name	Expression
Ackley	$f(x) = 20 \exp(0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 \cdot e$
Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{3000} - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$
Rastrigin	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$
Rosenbrock	$f(x) = \sum_{i=1}^n [100(x_i - x_{i-1}) + (x_i - 1)^2]$
Schwefel	$f(x) = 418.9829 \cdot n + \sum_{i=1}^n (x_i \cdot \sin(\sqrt{ x_i }))$
Exponential	$f(x) = -\exp(-0.5 \sum_{i=1}^n x_i^2) + 1$
Sphere	$f(x) = \sum_{i=1}^n x_i^2$
Parallel Ellipsoid	$f(x) = \sum_{i=1}^n (i \cdot x_i^2)$
Rotated Ellipsoid	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j^2)$

A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period 39

Table 1. (continued).

Function name	Search space [X_{max} , X_{min}]	Objective function value	Dim.	V_{max}
Ackley	$x \in [-32, 768, 32, 768]^*$	0	40	32,768
Griewank	$x \in [-300, 300]^*$	0	40	300
Rastrigin	$x \in [-5.12, 5.12]^*$	0	40	5.12
Rosenbrock	$x \in [-2, 048, 2, 048]^*$	0	40	2,048
Schwefel	$x \in [-500, 500]^*$	0	40	500
Exponential	$x \in [-1, 1]^*$	0	40	1
Sphere	$x \in [-5.12, 5.12]^*$	0	40	5.12
Parallel Ellipsoid	$x \in [-5.12, 5.12]^*$	0	40	5.12
Rotated Ellipsoid	$x \in [-65, 536, 65, 536]^*$	0	40	65,536

This research is conducted by a personal computer of Intel Core i5 2450 with a 2.5-GHz CPU and 8 GB RAM, and Visual C++ as the programming language. The measures of algorithm performance in the experiments are as follows: The mean best fitness value (MBF) is the mean of best fitness in the final iteration from all running (100 runs). MBF indicates the solution searching efficiency of an algorithm. All experiments benchmark functions have zero results as a minimum point. In Table 1, an entry less than 10^{-324} is given the value of zero. The closer the MBF to the zero point of a method, the better the method. The mean particles re-initialization period (MPRP) is the mean of particles re-initialization period in the final iteration from all running. MPRP is the result of the calculation of PSODR algorithm. The mean of time per mean of fitness (TPF) is the mean of time in running multiply the mean of fitness of GBEST in the final iteration from all running. The closer the TPF to the zero point of a method, the better the method. Because, this method loses a less time in the solution searching and obtains a good solution. SD is the standard deviation.

Table 2. The results of PSODR by fixed PRP on Benchmark Test Functions averaged over 100 runs.

Techniques	PSO	PSODR	PSODR	PSODR	PSODR	PSODR	PSODR
PRP	-	2	10	100	200	500	1000
Problems	MBF	MBF	MBF	MBF	MBF	MBF	MBF
Ackley	5.87489	5.20859	2.23955	2.16283	2.22361	2.89651	3.23259
Griewank	0.15754	1.59019	0.044938	0.0445354	0.048823	0.046841	0.052307
Rastrigin	158.724	332.885	135.183	139.055	145.385	146.352	150.278
Rosenbrock	9.67E-22	31.8709	2.10E-12	3.95E-24	4.89E-23	2.08E-20	1.37E-20
Schwefel	3979.11	3429.55	3143.13	3175.25	3102.46	3338.6	3614.64
Exponential	2.56E-15	0.0126073	2.92E-15	2.59E-16	2.24E-16	2.22E-16	2.45E-16
Sphere	3.66E-37	0.698723	2.13E-18	7.98E-35	2.43E-36	4.77E-37	2.04E-37
Parallel Ellipsoid	4.18E-36	30.2192	2.76E-14	5.29E-37	1.23E-36	3.28E-34	4.35E-37
Rotated Ellipsoid	2.42E-37	4651.77	2.94E-12	9.04E-35	1.20E-34	5.28E-37	2.30E-36

40 C. Ratanavilisagul and B. Kruatrachue

Table 3. Comparative results of PSO, PSOMEG, PSOMN, and PSODR on Benchmark Test Functions averaged over 100 runs.

Techniques Problems	PSO	PSOMEG	PSOMN	PSODR	
	MBF	MBF	MBF	MBF	MPRP
Ackley	5.87489	5.6928	3.2664	2.21957	103
Griewank	0.15754	0.133446	0.103264	0.04741	78
Rastrigin	158.724	153.939	148.766	138.981	67.8
Rosenbrock	9.67E-22	1.49E-19	2.60E-20	6.47E-23	192
Schwefel	3979.11	3910.7	3857.61	3091.22	66
Exponential	2.56E-15	1.15E-13	7.08E-14	2.71E-16	43.5
Sphere	3.66E-37	8.32E-36	1.51E-34	3.42E-37	200
Parallel Ellipsoid	4.18E-36	1.55E-34	1.04E-33	1.84E-36	200
Rotated Ellipsoid	2.42E-37	4.51E-36	9.08E-35	1.77E-37	200

Table 4. Comparative results of SD and TPF of PSO, PSOMEG, PSOMN, and PSODR on Benchmark Test Functions averaged over 100 runs.

Techniques Problems	PSO		PSOMEG		PSOMN		PSODR	
	SD	TPF	SD	TPF	SD	TPF	SD	TPF
Ackley	2.7838	1.2855	2.6532	1.3634	3.104	0.8522	0.5379	0.5029
Griewank	0.39	0.03535	0.1375	0.03749	0.2525	0.02883	0.04894	0.0108
Rastrigin	29.4346	31.4701	29.1122	33.143	31.5726	32.2524	25.87	29.3666
Rosenbrock	7.46E-13	1.54E-22	1.52E-20	2.48E-20	9.69E-15	4.24E-21	3.34E-21	1.03E-23
Schwefel	1063.64	1001.14	975.27	1006.73	955.22	1006.0646	933.454	791.661
Exponential	2.78E-14	3.67E-16	7.40E-14	1.66E-14	1.63E-12	1.02E-14	7.44E-17	3.91E-17
Sphere	1.78E-31	5.08E-38	1.59E-36	1.18E-36	6.44E-33	2.17E-35	3.47E-38	4.8E-38
Parallel Ellipsoid	1.41E-33	5.91E-37	2.64E-35	2.24E-35	2.37E-26	1.4823E-34	1.42E-36	2.62E-37
Rotated Ellipsoid	1.07E-35	5.53E-38	2.10E-35	1.03E-36	1.13E-28	2.0893E-35	3.20E-36	3.9205E-38

From the experimental results of Table 2 show both unimodal functions and multimodal functions, if the setting re-initialization period is 2, swarm will be over-reset. Its searching solution is poorer than that of the standard PSO, such as Griewank function, Rastrigin function, Rosenbrock function, Exponential function, Sphere function, Parallel Ellipsoid function, and Rotated Ellipsoid function. This is due to particles hardly converge. In the case of, re-initialization period is more than or equal to 100 in unimodal functions, overall its searching solutions are similar to that of the standard PSO since the particles re-initialization algorithms is not executed. For multimodal problems, if the setting re-initialization period is 1000, swarm will be under-reset. Its searching solution is poorer than setting re-initialization period to 100. Since swarm wastes time searching in local optimum rather than starting new search in other areas.

For multimodal problems in Table 2, PSODR gets good results because of the re-initialization period setting ranged from 10 to 200. The PSODR algorithm can adjust re-initialization period in this range and it can obtain the good solutions as

shown in Table 3. Therefore, the PSODR algorithm can dynamically adjust to suitably deal with the problem.

From the experimental results of Table 3 show that PSODR outperforms PSO, PSOMEG, and PSOMN in multimodal function. PSODR produces a better quality solution because of its lowest MBF in multimodal function. For unimodal function, PSO and PSODR have similar overall results. The MBF results of PSO and PSODR are higher than that of PSOMEG and PSOMN because PSOMEG and PSOMN could execute the particles re-initialization which will hinder the convergence of particles in unimodal function. For multimodal function, some MBF results of PSOMEG and PSOMN is poorer than the standard PSO such as Exponential function and Rosenbrock function. If adjusting re-initialization period to deal with problems appropriately, the solutions of PSOMEG and PSOMN can obtain better than that of the standard PSO. However, adjusting re-initialization period of PSOMEG and PSOMN to deal with any given problem appropriately is complicated.

From the experimental results of Table 4 show that the reliability of PSODR is better than that of PSO, PSOMEG, and PSOMN because of its lowest SD in all test functions. Moreover, PSODR obtains the best results when the results from the solution searching are compared with lost time in the solution searching. Because, TPF of PSODR is less than that of PSO, PSOMEG, and PSOMN in all test functions.

5 Conclusion

The particles re-initialization is applied with PSO to solve problems of trapping in local optimum and premature convergence. However, applying the particles re-initialization with general problems is complicated which is the cause of complicatedly adjusting the particle re-initialization period in order that the particles re-initialization can enhance the solution searching performance and can be easily applied to general problems. This paper proposes the re-initialization period that is automatically adjusted according to the difference of the best solution found after resetting and the previous best solution. The proposed technique is called PSODR. On benchmark functions, the proposed PSODR, PSO, PSOMEG and PSOMN are tested and the results are compared. From the experimental results show that PSODR can adjust the particle re-initialization period to solve the problem appropriately and PSODR outperforms PSO, PSOMEG, and PSOMN.

References

1. Kennedy, J., Eberhart, R. C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
2. Kennedy, J., Eberhart, R. C.: A New Optimizer Using Particle Swarm Theory. In: Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)

3. Paspoulopoulos, K. E., Plagianakos, V. P., Magoulas, G. D., Vralatis, M. N.: Objective function stretching to alleviate convergence to local minima, *Nonlinear Analysis TMA* 47, pp. 3419–3424 (2005)
4. Eberhart, R. C., Shi, Y.: comparison between genetic algorithms and particle swarm optimization. In: the 7th annual conference on evolutionary programming, San Diego (1998)
5. Kennedy, J., Eberhart, R. C.: *Swarm Intelligence*. Morgan Kaufmann, San Mateo (2001)
6. Pappala, V.S., Erlich, I., Rohrig, K., Dobschinski, J.A.: A stochastic model for the optimal operation of a wind-thermal power system. In: *IEEE Trans. Power Syst.*, pp. 940–950 (2009)
7. Acharjee, P., Goswami, S.K.: A decoupled power flow algorithm using particle swarm optimization technique, *Energy Convers. Manage.*, pp. 2351–2360 (2009)
8. Dutta, S., Singh, S.P.: Optimal rescheduling of generators for congestion management based on particle swarm optimization, *IEEE Trans. Power Syst.* (2008)
9. Kiranyaz, S., Ince, T., Yildirim, A., Gabbouj, M.: Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, *Neural Networks*, pp. 1448–1462 (2009)
10. Wei, H.L., Billings, S.A., Zhao, Y.F., Guo, L.Z.: Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification. In: *IEEE Trans. Neural Network*, pp. 181–185 (2009)
11. Lin, C.J., Chen, C.H., Lin, C.T.: A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications. In: *IEEE Trans. Syst. Man Cybernetics Part C*, pp. 55–68 (2009)
12. Zhao, L., Qian, F., Yang, Y., Zeng, Y., Su, H.: Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization. In: *Appl. Soft Comput.*, pp. 938–944 (2010)
13. Van Den Bergh, F.: An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria, South Africa (2002)
14. Li, N., Suan, D., Cen, Y., Zou, T.: Particle swarm optimization with mutation operator. In: *Computer Engineering and Applications*, vol. 17, pp. 12–14 (2004)
15. Liu, X., Wang, Q., Liu, H., Li, L.: Particle Swarm Optimization with Dynamic Inertia Weight and Mutation. In: *Third International Conference on Genetic and Evolutionary Computing*, pp. 620–623 (2009)
16. Chabwood, R., Boontee, K.: Mutation Period Calculation for Particle Swarm Optimization. In: *1st International Symposium on Technology for Sustainability*, pp. 213–216 (2011)
17. Andrews, P. S.: An investigation into mutation operators for particle swarm optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, Canada: IEEE, pp. 1044–1051 (2006)
18. Lin, M., Hua, Z.: Improved PSO Algorithm with Adaptive Inertia Weight and Mutation. In: *2009 World Congress on Computer Science and Information Engineering*, pp. 622–625 (2009)
19. Higashi, N., Iba, H.: Particle swarm optimization with Gaussian mutation. In: *Proc. of the 2003 IEEE Swarm Intelligence Symposium*, pp. 72–79 (2003)

ภาคผนวก ซ

การทดลองพารามิเตอร์ของการกลายพันธุ์และรีโพซิชัน

ในภาคผนวกนี้จะดำเนินการทดลองเพื่อแสดงผลของการเลือกพารามิเตอร์ของขั้นตอนวิธีที่นำเสนอ (MRPSO) ที่ทดลองกับฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ และปัญหาการเดินทางของพนักงานขาย โดยพารามิเตอร์เหล่านี้ถูกใช้ในการทดลองในบทที่ 5 และภาคผนวก ง และ จ

สำหรับพารามิเตอร์ของ MRPSO ที่เพิ่มขึ้นมาจาก PSO ปกติ คือรอบของการกลายพันธุ์ (Round of Mutation หรือ RM) ความน่าจะเป็นในการกลายพันธุ์ (Probability of Mutation หรือ PM) ค่าที่กำหนดรอบของรีโพซิชัน (Threshold of Reposition หรือ TR) และความน่าจะเป็นในการรีโพซิชัน (Probability of Reposition หรือ PR) โดยการทดลองจะทำการแยกพารามิเตอร์ออกเป็นสองชุดเพื่อให้ง่ายต่อการวิเคราะห์ สำหรับชุดแรกทำการทดลองปรับพารามิเตอร์ RM และ PM โดยจะทำการทดลองกับ MXUPG ซึ่งเป็นส่วนประกอบของ MRPSO สำหรับชุดที่สองทำการทดลองปรับพารามิเตอร์ TR และ PR โดยจะทำการทดลองกับ RPG ซึ่งเป็นส่วนประกอบของ MRPSO

สำหรับรายละเอียดขั้นตอนวิธีที่นำมาทดลองถูกกล่าวไว้ในบทที่ 5 และรายละเอียดของฟังก์ชันมาตรฐาน ปัญหาถุงเป้หลายมิติ และปัญหาการเดินทางของพนักงานขายที่นำมาทดลองถูกกล่าวไว้ในบทที่ 5 ส่วนการทดลองสามารถแบ่งออกเป็น 3 ส่วน โดยพิจารณาจากจุดประสงค์การทดลองมีรายละเอียดดังนี้

- การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG โดยทดลองกับฟังก์ชันมาตรฐาน
- การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG โดยทดลองกับปัญหาถุงเป้หลายมิติ
- การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG โดยทดลองกับปัญหาการเดินทางของพนักงานขาย

ซ.1 การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG กับฟังก์ชันมาตรฐาน

ในหัวข้อนี้วิเคราะห์ผลกระทบจากปรับพารามิเตอร์ RM และ PM ของ MXUPG และพารามิเตอร์ TR และ PR ของ RPG ที่ทดลองกับฟังก์ชันมาตรฐาน

โดยตาราง ซ.1 ได้ใช้พารามิเตอร์ตามตาราง 5.1 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ RM จึงกำหนด PM ให้มีค่าคงที่โดยกำหนดตามตาราง 5.1 แต่สำหรับจำนวนประชากรและ RM ถูกกำหนดตามตาราง ซ.1

โดยตาราง ซ.2 ได้ใช้พารามิเตอร์ตามตาราง 5.1 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ PM จึงกำหนด RM ให้มีค่าคงที่โดยกำหนดตามตาราง 5.1 แต่สำหรับ PM ถูกกำหนดตามตาราง ซ.2

โดยตาราง ข.3 ได้ใช้พารามิเตอร์ตามตาราง 5.1 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ TR จึงกำหนด PR ให้มีค่าคงที่โดยกำหนดตามตาราง 5.1 แต่สำหรับ TR ถูกกำหนดตามตาราง ข.3

โดยตาราง ข.4 ได้ใช้พารามิเตอร์ตามตาราง 5.1 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ PR จึงกำหนด TR ให้มีค่าคงที่โดยกำหนดตามตาราง 5.1 แต่สำหรับ PR ถูกกำหนดตามตาราง ข.4 โดยตาราง ข.1 ถึง ข.4 แสดงค่า MBF และ SR

ตารางที่ ข.1 ผลการค้นหาคำตอบการปรับค่า RM กับฟังก์ชันมาตรฐานด้วย MXUPG

Algorithm	40 อนุภาค RM 5		20 อนุภาค RM 10		10 อนุภาค RM 20	
	MBF	SR	MBF	SR	MBF	SR
ACKLEY	8.95E-15	0	9.59E-15	0	1.01E-14	0
GRIEWANK	0.00147808	86	0.00192444	81	0.000936307	90
RASTRIGIN	2.10E-14	73	1.88E-14	75	1.93E-14	80
ROSENBROCK	2.11E-28	4	3.92E-27	0	3.92E-26	0
SCHWEFEL	2981.63	0	2643.34	0	2312.06	0
COSINE	0	100	0	100	0	100
EXPONENTIAL	1.13E-16	0	1.14E-16	0	1.19E-16	0
LEVY	0.14414	5	0.248889	2	0.481662	0
MICHALEWICZ	0.370511	67	0.26702	75	0.123104	87
STEP	0	100	0	100	0	100
SCHAFFER	0	100	0	100	0.000194318	98
HOLDER	0	100	0	100	3.16E-06	82
BEALE	0	100	0	100	0.022863	97
SHUBERT	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100
TRID	9.94E-12	0	1.67E-11	0	2.64E-08	0
SPHERE	4.22E-120	0	5.48E-119	0	2.73E-119	0
PARALLEL	4.48E-118	0	7.38E-118	0	2.29E-117	0
ROTATED	6.34E-118	0	1.32E-118	0	7.97E-116	0
CIGAR	3.08E-113	0	1.64E-111	0	5.81E-113	0
BROWN	7.95E-119	0	1.55E-119	0	1.66E-119	0
MULTIMOD	1.68E-60	0	1.83E-59	0	4.34E-66	0
ZAKHAROV	4.76E-32	0	5.72E-39	0	4.69E-48	0
EASOM	0	100	0	100	0	100
ผลรวม		1035		1033		1034

จากตาราง ข.1 โดยภาพรวมแล้วการปรับพารามิเตอร์ RM ของ MXUPG ได้รับผลลัพธ์การค้นหาใกล้เคียงกัน ดังนั้นการเลือกพารามิเตอร์ RM ของ MXUPG สามารถเลือกใช้ค่าใดก็ได้ สำหรับวิทยานิพนธ์ฉบับนี้ ได้เลือกใช้ค่า RM เท่ากับ 10 และกำหนดจำนวนประชากรเท่ากับ 20 อนุภาค

จากตาราง ข.2 โดยภาพรวมแล้วการปรับพารามิเตอร์ PM ของ MXUPG ได้รับผลลัพธ์การค้นหาที่แตกต่างกันเล็กน้อย โดยการกำหนดค่า PM เท่ากับ 0.05 จะได้รับคำตอบที่ดีที่สุด ดังนั้นวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า PM เท่ากับ 0.05

ตารางที่ ข.2 ผลการค้นหาคำตอบการปรับค่า PM กับฟังก์ชันมาตรฐานด้วย MXUPG

Algorithm	PM 0.01		PM 0.03		PM 0.05		PM 0.07	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	1.42E-14	0	1.14E-14	0	9.59E-15	0	8.17E-15	0
GRIEWANK	0.00610511	57	0.00332488	71	0.00192444	81	0.00322178	75
RASTRIGIN	1.04E-13	15	4.15E-14	54	1.88E-14	75	2.33E-14	74
ROSENBROCK	9.58E-25	0	1.09E-24	0	3.92E-27	0	7.94E-24	0
SCHWEFEL	2750.34	0	2545.84	0	2643.34	0	2613.36	0
COSINE	3.11E-17	96	0	100	0	100	0	100
EXPONENTIAL	1.65E-16	0	1.21E-16	0	1.14E-16	0	1.11E-16	0
LEVY	0.247098	3	0.26948	2	0.248889	2	0.26948	1
MICHALEWICZ	0.267828	73	0.229049	80	0.26702	75	0.236642	76
STEP	0	100	0	100	0	100	0	100
SCHAFFER	0.0016517	83	0.00048579	95	0	100	0	100
HOLDER	7.62E-08	97	2.97E-07	96	0	100	0	100
BEALE	0	100	0.0381037	95	0	100	0.0152415	98
SHUBERT	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100
TRID	1.56E-11	0	1.63E-11	0	1.67E-11	0	1.64E-11	0
SPHERE	6.05E-97	0	5.09E-112	0	5.48E-119	0	6.10E-124	0
PARALLEL	4.03E-96	0	3.42E-111	0	7.38E-118	0	1.24E-123	0
ROTATED	3.15E-95	0	1.94E-111	0	1.32E-118	0	3.57E-122	0
CIGAR	3.03E-92	0	3.47E-107	0	1.64E-111	0	5.72E-119	0
BROWN	1.41E-98	0	2.65E-113	0	1.55E-119	0	3.52E-124	0
MULTIMOD	1.65E-51	0	4.08E-58	0	1.83E-59	0	2.48E-69	0
ZAKHAROV	1.81E-20	0	1.74E-32	0	5.72E-39	0	3.22E-43	0
EASOM	0	100	0	100	0	100	0	100
ผลรวม		924		993		1033		1024

จากตาราง ข.3 การกำหนดค่า TR ของ RPG เท่ากับ 100 จะได้รับคำตอบที่ดีที่สุด โดยการกำหนดค่า TR ให้มีค่าน้อยเกินไป ($TR = 50$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาย่ำแย่ลงกว่าการกำหนดค่า TR ให้เหมาะสมกับการแก้ปัญหา และในทางกลับกันการกำหนดค่า TR ให้มีค่ามากเกินไป ($TR = 500$ และ 1000) อาจจะทำให้เกิดปัญหาการกลายพันธุ์น้อยเกินไปทำให้ผลลัพธ์การค้นหาย่ำแย่กว่าการกำหนดค่า TR ให้เหมาะสมกับการแก้ปัญหา ดังนั้นวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า TR เท่ากับ 100

จากตาราง ข.4 การกำหนดค่า PR ของ RPG เท่ากับ 0.3 จะได้รับคำตอบที่ดีที่สุด โดยการกำหนดค่า PR ให้มีค่าน้อยเกินไป ($PR = 0.1$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์น้อยเกินไปทำให้ผลลัพธ์การค้นหาย่ำแย่กว่าการกำหนดค่า PR ให้เหมาะสมกับการแก้ปัญหา และในทางกลับกันการกำหนดค่า PR ให้มีค่ามากเกินไป ($PR = 0.5$ และ 0.7) อาจจะทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาย่ำแย่กว่าการกำหนดค่า PR ให้เหมาะสมกับการแก้ปัญหาดังนั้นวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า PR เท่ากับ 0.3

ตารางที่ ข.3 ผลการค้นหาคำตอบการปรับค่า TR กับฟังก์ชันมาตรฐานด้วย RPG

Algorithm	TR 50		TR 100		TR 500		TR 1000	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	3.20E-16	91	1.07E-16	97	6.29E-15	0	7.28E-15	0
GRIEWANK	0.000838	90	0	100	0.002956	72	0.003902	57
RASTRIGIN	0.18915	72	0.368405	73	10.7274	0	18.4593	0
ROSENBROCK	0	100	0	100	0	100	0	100
SCHWEFEL	1852.86	0	1769.35	0	1910.73	0	2054.65	0
COSINE	0	100	0	100	0	100	0	100
EXPONENTIAL	0	100	0	100	2.00E-17	82	8.22E-17	26
LEVY	0.000895	96	0	100	0.006267	92	0.005372	90
MICHALEWICZ	0.09239	90	0.080718	92	0.091292	77	0.136158	51
STEP	0	100	0	100	0	100	0	100
SCHAFFER	0.000291	97	9.72E-05	99	0	100	9.72E-05	99
HOLDER	0	100	0	100	0	100	0	100
BEALE	0	100	0	100	0	100	0	100
SHUBERT	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100
TRID	2.73E-13	79	1.82E-13	84	2.70E-12	23	3.27E-12	22
SPHERE	8.76E-118	0	4.75E-118	0	5.85E-118	0	2.48E-117	0
PARALLEL	2.28E-117	0	1.16E-116	0	6.72E-117	0	1.74E-116	0
ROTATED	6.13E-118	0	3.55E-117	0	1.66E-117	0	3.91E-117	0
CIGAR	2.18E-112	0	2.52E-112	0	6.75E-112	0	7.80E-113	0
BROWN	2.05E-118	0	6.87E-119	0	8.98E-119	0	2.22E-118	0
MULTIMOD	3.35E-48	0	1.37E-55	0	1.43E-13	0	6.61682	0
ZAKHAROV	7.73E-16	0	9.64E-16	0	1.27E-15	0	1.10E-15	0
EASOM	0	100	0	100	0	100	0	100
ผลรวม		1515		1545		1246		1145

ตารางที่ ข.4 ผลการค้นหาคำตอบการปรับค่า PR กับฟังก์ชันมาตรฐานด้วย RPG

Algorithm	PR 0.1		PR 0.3		PR 0.5		PR 0.7	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
ACKLEY	5.51E-15	0	1.07E-16	97	1.03E-15	71	3.38E-15	5
GRIEWANK	0.004358	67	0	100	0	100	0.001134	86
RASTRIGIN	1.97229	17	0.368405	73	1.07501	50	1.48532	38
ROSENBROCK	0	100	0	100	0	100	0	100
SCHWEFEL	1811.41	0	1769.35	0	1947.23	0	1920.89	0
COSINE	0	100	0	100	0	100	0	100
EXPONENTIAL	1.55E-17	86	0	100	0	100	0	100
LEVY	0.001791	96	0	100	0	100	0	100
MICHALEWICZ	0.035299	96	0.080718	92	0.0454925	70	0.080089	37
STEP	0	100	0	100	0	100	0	100
SCHAFFER	0	100	9.72E-05	99	0	100	0	100
HOLDER	0	100	0	100	0	100	0	100
BEALE	0	100	0	100	0	100	0	100
SHUBERT	0	100	0	100	0	100	0	100
GOLDSTEIN	0	100	0	100	0	100	0	100
TRID	2.01E-12	24	1.82E-13	84	4.37E-13	67	7.73E-13	53
SPHERE	1.29E-117	0	4.75E-118	0	1.35E-118	0	1.36E-117	0
PARALLEL	1.18E-116	0	1.16E-116	0	1.61E-115	0	1.14E-116	0
ROTATED	4.42E-117	0	3.55E-117	0	5.15E-117	0	2.42E-118	0
CIGAR	4.37E-112	0	2.52E-112	0	3.73E-111	0	2.84E-112	0
BROWN	3.59E-119	0	6.87E-119	0	1.70E-118	0	2.22E-119	0
MULTIMOD	1.79E-21	0	1.37E-55	0	4.95E-54	0	4.63E-62	0
ZAKHAROV	9.52E-16	0	9.64E-16	0	1.06E-15	0	2.63E-15	0
EASOM	0	100	0	100	0	100	0	100
ผลรวม		1286		1545		1458		1319

ข.2 การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG กับ MKP

ในหัวข้อนี้วิเคราะห์ผลกระทบจากปรับพารามิเตอร์ RM และ PM ของ MXUPG และ พารามิเตอร์ TR และ PR ของ RPG ที่ทดลองกับปัญหาถุงเป้แบบหลายมิติ

โดยตาราง ข.5 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่รอบของการค้นหาคำตอบ 1000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ RM จึงกำหนด PM ให้มีค่าคงที่โดยกำหนดตามตาราง 5.2 แต่สำหรับจำนวนประชากรและ RM ถูกกำหนดตามตาราง ข.5

โดยตาราง ข.6 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่รอบของการค้นหาคำตอบ 1000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ PM จึงกำหนด RM ให้มีค่าคงที่โดยกำหนดตามตาราง 5.2 แต่สำหรับ PM ถูกกำหนดตามตาราง ข.6

ตารางที่ ๕.5 ผลการค้นหาคำตอบการปรับค่า RM กับ MKP ด้วย MXUPG

Algorithm	100 อนุภาค RM 10		200 อนุภาค RM 5		500 อนุภาค RM 1	
	MBF	SR	MBF	SR	MBF	SR
SENTO1	7767.39	78	7768.43	83	7768.41	78
SENTO2	8715.06	41	8716.59	52	8715.87	48
WEING1	141278	100	141278	100	141278	100
WEING2	130883	100	130883	100	130883	100
WEING3	95271.3	22	95383.7	31	95573.2	65
WEING4	119337	100	119337	100	119337	100
WEING5	98147	83	98569.3	94	98758.2	99
WEING6	130575	90	130607	96	130607	96
WEING7	1095240	4	1095220	1	1095200	3
WEING8	620261	1	620422	6	620572	6
WEISH01	4554	100	4554	100	4554	100
WEISH02	4535.3	86	4535	80	4535.3	86
WEISH03	4113.74	98	4114.37	99	4115	100
WEISH04	4561	100	4561	100	4561	100
WEISH05	4514	100	4514	100	4514	100
WEISH06	5553.94	83	5553.6	80	5553.68	81
WEISH07	5566.57	98	5567	100	5567	100
WEISH08	5604.92	96	5604.9	95	5605	100
WEISH09	5246	100	5246	100	5246	100
WEISH10	6339	100	6339	100	6339	100
WEISH11	5642.96	99	5642.92	98	5643	100
WEISH12	6339	100	6339	100	6339	100
WEISH13	6159	100	6159	100	6159	100
WEISH14	6954	100	6954	100	6954	100
WEISH15	7486	100	7486	100	7486	100
WEISH16	7288.16	58	7288.35	67	7288.6	80
WEISH17	8633	100	8633	100	8633	100
WEISH18	9578.17	82	9579.17	93	9579.49	95
WEISH19	7693.32	64	7695.66	82	7695.53	81
WEISH20	9449.8	99	9450	100	9449.8	99
WEISH21	9063.55	70	9065.77	73	9072.55	95
WEISH22	8932.78	21	8936.02	39	8936.02	39
WEISH23	8343.1	70	8343.55	85	8343.73	91
WEISH24	10219.5	97	10219.8	99	10219.6	98
WEISH25	9932.83	62	9932.35	59	9934.28	71
WEISH26	9560.58	26	9562.88	34	9567.8	50
WEISH27	9819	100	9819	100	9819	100
WEISH28	9492	100	9492	100	9492	100
WEISH29	9408.56	95	9410	100	9410	100
WEISH30	11190.5	93	11190.6	91	11190.6	91
PB1	3077.05	47	3079.88	55	3082.84	68
PB2	3179.81	77	3181.52	81	3181.73	81
PB4	94929.1	93	95103.3	98	95168	100
PB5	2127.95	47	2130.51	59	2133.07	71
PB6	774.43	90	775.3	95	775.58	97
PB7	1031.85	67	1032.46	73	1033.23	81
HP1	3402.41	32	3405.7	47	3407.1	48
HP2	3180.44	77	3181.19	78	3182.86	82
ผลรวม		3746		3923		4080

ตารางที่ ข.6 ผลการค้นหาค่าตอบการปรับค่า PM กับ MKP ด้วย MXUPG

Algorithm	PM 0.01		PM 0.03		PM 0.05		PM 0.07	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
SENTO1	7768.72	79	7768.74	84	7768.41	78	7767.11	76
SENTO2	8713.94	32	8716.81	50	8715.87	48	8715.7	34
WEING1	141278	100	141278	100	141278	100	141278	100
WEING2	130883	100	130883	100	130883	100	130883	100
WEING3	95572.7	64	95563.3	74	95573.2	65	95589.1	68
WEING4	119337	100	119337	100	119337	100	119337	100
WEING5	98796	100	98796	100	98758.2	99	98720.4	98
WEING6	130565	85	130590	94	130607	96	130611	97
WEING7	1095230	8	1095260	4	1095200	3	1095240	5
WEING8	620934	12	620378	1	620572	6	620683	5
WEISH01	4554	100	4554	100	4554	100	4554	100
WEISH02	4534.95	79	4535.35	87	4535.3	86	4535.15	83
WEISH03	4115	100	4115	100	4115	100	4115	100
WEISH04	4561	100	4561	100	4561	100	4561	100
WEISH05	4514	100	4514	100	4514	100	4514	100
WEISH06	5552.31	72	5553.75	80	5553.68	81	5553.03	76
WEISH07	5566.21	96	5567	100	5567	100	5567	100
WEISH08	5604.67	89	5605	100	5605	100	5604.94	97
WEISH09	5246	100	5246	100	5246	100	5246	100
WEISH10	6339	100	6339	100	6339	100	6339	100
WEISH11	5642.92	98	5642.92	98	5643	100	5642.84	96
WEISH12	6339	100	6339	100	6339	100	6339	100
WEISH13	6159	100	6159	100	6159	100	6159	100
WEISH14	6954	100	6954	100	6954	100	6954	100
WEISH15	7486	100	7486	100	7486	100	7486	100
WEISH16	7288.58	87	7288.68	84	7288.6	80	7288.68	84
WEISH17	8633	100	8633	100	8633	100	8633	100
WEISH18	9578.04	80	9578.61	86	9579.49	95	9578.37	86
WEISH19	7695.01	77	7696.05	85	7695.53	81	7695.79	83
WEISH20	9449.2	96	9449.4	97	9449.8	99	9449.6	98
WEISH21	9069.63	87	9072.31	94	9072.55	95	9069.86	89
WEISH22	8935.9	41	8936.56	42	8936.02	39	8937.64	48
WEISH23	8342.23	74	8343.19	84	8343.73	91	8342.8	82
WEISH24	10219.5	97	10219.8	99	10219.6	98	10218.5	94
WEISH25	9933.62	67	9933.39	65	9934.28	71	9932.37	61
WEISH26	9573.36	68	9568.24	52	9567.8	50	9573.04	67
WEISH27	9819	100	9819	100	9819	100	9819	100
WEISH28	9492	100	9492	100	9492	100	9492	100
WEISH29	9407.6	94	9409.68	99	9410	100	9408.4	95
WEISH30	11189.8	71	11190.4	86	11190.6	91	11190.2	80
PB1	3081.67	64	3080.44	57	3082.84	68	3082.4	63
PB2	3170.06	60	3173.98	59	3181.73	81	3175.41	59
PB4	95164.1	99	95164.1	99	95168	100	95164.1	99
PB5	2131.21	70	2133.19	68	2133.07	71	2130.56	63
PB6	775.16	94	774.74	91	775.58	97	775.44	96
PB7	1033.24	80	1033.63	85	1033.23	81	1033.21	78
HP1	3404.54	45	3406.6	52	3407.1	48	3407.56	55
HP2	3173.13	63	3181.27	79	3182.86	82	3177.2	73
ผลรวม		3928		4035		4080		3988

ตารางที่ ข.7 ผลการค้นหาคำตอบการปรับค่า TR กับ MKP ด้วย RPG

Algorithm	TR 50		TR 100		TR 500	
	MBF	SR	MBF	SR	MBF	SR
SENTO1	7769.62	83	7769.48	82	7769.25	84
SENTO2	8719.64	71	8719.77	73	8716.23	40
WEING1	141278	100	141278	100	141278	100
WEING2	130883	100	130883	100	130883	100
WEING3	95677	100	95677	100	95573.7	66
WEING4	119337	100	119337	100	119337	100
WEING5	98796	100	98796	100	98796	100
WEING6	130623	100	130623	100	130619	99
WEING7	1095410	50	1095420	61	1095260	5
WEING8	621606	19	622091	32	620446	2
WEISH01	4554	100	4554	100	4554	100
WEISH02	4535.05	81	4535.45	89	4535.2	84
WEISH03	4115	100	4115	100	4115	100
WEISH04	4561	100	4561	100	4561	100
WEISH05	4514	100	4514	100	4514	100
WEISH06	5556.69	98	5556.69	98	5554.34	83
WEISH07	5567	100	5567	100	5567	100
WEISH08	5605	100	5605	100	5605	100
WEISH09	5246	100	5246	100	5246	100
WEISH10	6339	100	6339	100	6339	100
WEISH11	5643	100	5643	100	5642.96	99
WEISH12	6339	100	6339	100	6339	100
WEISH13	6159	100	6159	100	6159	100
WEISH14	6954	100	6954	100	6954	100
WEISH15	7486	100	7486	100	7486	100
WEISH16	7289	100	7289	100	7288.8	90
WEISH17	8633	100	8633	100	8633	100
WEISH18	9580	100	9580	100	9578.49	83
WEISH19	7697.74	98	7697.87	99	7696.7	90
WEISH20	9450	100	9450	100	9449.8	99
WEISH21	9074	100	9074	100	9071.09	91
WEISH22	8942.14	73	8943.76	82	8936.2	40
WEISH23	8344	100	8344	100	8343.61	87
WEISH24	10220	100	10220	100	10220	100
WEISH25	9937.88	93	9938.84	99	9932.1	57
WEISH26	9582.37	94	9581.76	93	9568.44	52
WEISH27	9819	100	9819	100	9819	100
WEISH28	9492	100	9492	100	9492	100
WEISH29	9410	100	9410	100	9409.04	97
WEISH30	11190.8	96	11191	99	11190.4	85
PB1	3087.26	85	3086.78	77	3084.4	69
PB2	3182.17	83	3184.56	92	3176.74	67
PB4	95168	100	95168	100	95168	100
PB5	2137.3	90	2138.83	99	2132.07	67
PB6	776	100	776	100	775.44	96
PB7	1034.28	92	1034.01	89	1033.13	78
HP1	3412.76	66	3415.06	79	3406.26	52
HP2	3184.38	91	3185.1	95	3181.66	82
ผลรวม		4463		4538		4044

ตารางที่ ๘.8 ผลการค้นหาคำตอบการปรับค่า PR กับ MKP ด้วย RPG

Algorithm	PR 0.1		PR 0.3		PR 0.5		PR 0.7	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
SENTO1	7768.75	78	7769.48	82	7769.06	79	7766.42	65
SENTO2	8716.99	50	8719.77	73	8719.64	69	8715.41	37
WEING1	141278	100	141278	100	141278	100	141278	100
WEING2	130883	100	130883	100	130883	100	130883	100
WEING3	95669.3	99	95677	100	95677	100	95669.3	99
WEING4	119337	100	119337	100	119337	100	119337	100
WEING5	98796	100	98796	100	98796	100	98796	100
WEING6	130623	100	130623	100	130623	100	130623	100
WEING7	1095350	12	1095420	61	1095410	45	1095390	29
WEING8	620983	10	622091	32	621833	27	620844	9
WEISH01	4554	100	4554	100	4554	100	4554	100
WEISH02	4535.35	87	4535.45	89	4535.65	93	4535.2	84
WEISH03	4115	100	4115	100	4115	100	4115	100
WEISH04	4561	100	4561	100	4561	100	4561	100
WEISH05	4514	100	4514	100	4514	100	4514	100
WEISH06	5555.14	88	5556.69	98	5556.07	94	5553.46	77
WEISH07	5567	100	5567	100	5567	100	5567	100
WEISH08	5605	100	5605	100	5605	100	5605	100
WEISH09	5246	100	5246	100	5246	100	5246	100
WEISH10	6339	100	6339	100	6339	100	6339	100
WEISH11	5643	100	5643	100	5643	100	5643	100
WEISH12	6339	100	6339	100	6339	100	6339	100
WEISH13	6159	100	6159	100	6159	100	6159	100
WEISH14	6954	100	6954	100	6954	100	6954	100
WEISH15	7486	100	7486	100	7486	100	7486	100
WEISH16	7288.88	97	7289	100	7289	100	7288.7	96
WEISH17	8633	100	8633	100	8633	100	8633	100
WEISH18	9579.49	95	9580	100	9580	100	9579.71	97
WEISH19	7697.22	94	7697.87	99	7697.87	99	7697.35	95
WEISH20	9450	100	9450	100	9450	100	9449.8	99
WEISH21	9074	100	9074	100	9074	100	9074	100
WEISH22	8939.62	59	8943.76	82	8942.68	76	8937.64	48
WEISH23	8343.97	99	8344	100	8344	100	8343.61	87
WEISH24	10219.8	99	10220	100	10220	100	10220	100
WEISH25	9935.48	78	9938.84	99	9938.84	99	9935.87	81
WEISH26	9578.04	82	9581.76	93	9581.98	94	9576.56	78
WEISH27	9819	100	9819	100	9819	100	9819	100
WEISH28	9492	100	9492	100	9492	100	9492	100
WEISH29	9410	100	9410	100	9410	100	9409.36	98
WEISH30	11190	75	11191	99	11191	99	11190.6	89
PB1	3085.31	71	3086.78	77	3086.92	78	3085.8	78
PB2	3179.26	72	3184.56	92	3183.01	90	3175.13	73
PB4	95168	100	95168	100	95168	100	95168	100
PB5	2136.79	87	2138.83	99	2137.81	93	2135.6	80
PB6	776	100	776	100	776	100	776	100
PB7	1033.71	85	1034.01	89	1034.55	95	1033.57	85
HP1	3411.56	68	3415.06	79	3413.6	72	3411.88	68
HP2	3182.96	91	3185.1	95	3184.2	90	3180.6	86
ผลรวม		4276		4538		4492		4238

โดยตาราง ซ.7 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่รอบของการค้นหาคำตอบ 1000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ TR จึงกำหนด PR ให้มีค่าคงที่โดยกำหนดตามตาราง 5.3 แต่สำหรับ TR ถูกกำหนดตามตาราง ซ.7

โดยตาราง ซ.8 ได้ใช้พารามิเตอร์ตามตาราง 5.2 แต่รอบของการค้นหาคำตอบ 1000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ PR จึงกำหนด TR ให้มีค่าคงที่โดยกำหนดตามตาราง 5.2 แต่สำหรับ PR ถูกกำหนดตามตาราง ซ.8 โดยตาราง ซ.5 ถึง ซ.8 แสดงค่า MBF และ SR

จากตาราง ซ.5 การกำหนดค่า RM ของ MXUPG เท่ากับ 1 และมีจำนวนอนุภาค 500 จะได้รับคำตอบที่ดีที่สุด โดยจากตาราง พบว่าการกำหนดค่า RM เท่ากับ 10 และมีจำนวนอนุภาค 100 ได้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า RM เท่ากับ 5 และมีจำนวนอนุภาค 200 สำหรับการกำหนดค่า RM เท่ากับ 5 และมีจำนวนอนุภาค 200 ได้ผลลัพธ์การค้นหาคำตอบแย่กว่า RM เท่ากับ 2 และมีจำนวนอนุภาค 500 ดังนั้นการกำหนดให้มีการกลายพันธุ์น้อยและให้มีจำนวนอนุภาค มากๆ จะได้รับผลลัพธ์การค้นหาดีกว่าการกำหนดให้มีจำนวนอนุภาคน้อยแต่ให้ดำเนินการกลายพันธุ์ มาก สำหรับกรณี MKP ควรกำหนดให้มีจำนวนการกลายพันธุ์น้อย แต่มีจำนวนอนุภาคมาก ดังนั้น วิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า RM เท่ากับ 1 และมีจำนวนอนุภาค 500

จากตาราง ซ.6 โดยภาพรวมแล้วการปรับพารามิเตอร์ PM ของ MXUPG ให้มีค่าเท่ากับ 0.03 และ 0.05 ให้ผลลัพธ์แตกต่างกันไม่มาก ซึ่งถ้าเลือกพารามิเตอร์ในช่วงเหล่านี้ก็จะได้รับคำตอบที่ไม่แตกต่างกันมากนัก แต่จากผลการทดลองการกำหนดค่า PM เท่ากับ 0.05 จะได้รับคำตอบที่ดีที่สุด ดังนั้น วิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า PM เท่ากับ 0.05

จากตาราง ซ.7 การกำหนดค่า TR ของ RPG เท่ากับ 100 จะได้รับคำตอบที่ดีที่สุด โดยการ กำหนดค่า TR ให้มีค่าน้อยเกินไป ($TR = 50$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวาง กระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาแย่กว่าการกำหนดค่า TR ให้เหมาะสมกับการแก้ปัญหา และในทางกลับกันการกำหนดค่า TR ให้มีค่ามากเกินไป ($TR = 500$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์น้อยเกินไปทำให้ผลลัพธ์การค้นหาแย่กว่าการกำหนดค่า TR ให้เหมาะสมกับการแก้ปัญหา ดังนั้น วิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า TR เท่ากับ 100

จากตาราง ซ.8 การกำหนดค่า PR ของ RPG เท่ากับ 0.3 จะได้รับคำตอบที่ดีที่สุด โดยการ กำหนดค่า PR ให้มีค่าน้อยเกินไป ($PR = 0.1$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์น้อยเกินไปทำให้ ผลลัพธ์การค้นหาแย่กว่าการกำหนดค่า PR ให้เหมาะสมกับการแก้ปัญหา และในทางกลับกันการ กำหนดค่า PR ให้มีค่ามากเกินไป ($PR = 0.5$ และ 0.7) อาจจะทำให้เกิดปัญหาการกลายพันธุ์ที่ ขัดขวางกระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาแย่กว่าการกำหนดค่า PR ให้เหมาะสมกับการ แก้ปัญหา ดังนั้นวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า PR เท่ากับ 0.3

ข.3 การทดลองการปรับพารามิเตอร์ของ MXUPG และ RPG กับ TSP

ในหัวข้อนี้วิเคราะห์ผลกระทบจากปรับพารามิเตอร์ RM และ PM ของ MXUPG และ พารามิเตอร์ TR และ PR ของ RPG ที่ทดลองกับปัญหาการเดินทางของพนักงานขาย

โดยตาราง ข.9 ได้ใช้พารามิเตอร์ตามตาราง 5.3 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ RM จึงกำหนด PM ให้มีค่าคงที่โดยกำหนดตามตาราง 5.3 แต่สำหรับจำนวนประชากรและ RM ถูกกำหนดตามตาราง ข.9

โดยตาราง ข.10 ได้ใช้พารามิเตอร์ตามตาราง 5.3 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ PM จึงกำหนด RM ให้มีค่าคงที่โดยกำหนดตามตาราง 5.3 แต่สำหรับ PM ถูกกำหนดตามตาราง ข.10

โดยตาราง ข.11 ได้ใช้พารามิเตอร์ตามตาราง 5.3 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ TR จึงกำหนด PR ให้มีค่าคงที่โดยกำหนดตามตาราง 5.3 แต่สำหรับ TR ถูกกำหนดตามตาราง ข.11

โดยตาราง ข.12 ได้ใช้พารามิเตอร์ตามตาราง 5.3 แต่รอบของการค้นหาคำตอบ 5000 รอบ เนื่องจากต้องการแสดงผลการปรับพารามิเตอร์ PR จึงกำหนด TR ให้มีค่าคงที่โดยกำหนดตามตาราง 5.3 แต่สำหรับ PR ถูกกำหนดตามตาราง ข.12 โดยตาราง ข.9 ถึง ข.12 แสดงค่า MBF และ SR

ตารางที่ ข.9 ผลการค้นหาคำตอบการปรับค่า RM กับ TSP ด้วย MXUPG

Algorithm	10 อนุภาค RM 40		20 อนุภาค RM 20		40 อนุภาค RM 10	
	MBF	SR	MBF	SR	MBF	SR
BURMA14	3323.39	97	3323	100	3323	100
ULYSSES16	6866.47	57	6861.71	79	6862.52	73
GR17	2087.36	80	2085.58	88	2085.3	94
GR21	2740.41	77	2707	100	2714.68	96
ULYSSES22	7079.22	27	7031.68	78	7056.99	47
FRI26	962.75	15	952.85	25	951.73	32
BAYG29	1684.32	4	1639.13	32	1646.46	23
BAYS29	2075.77	4	2040	18	2046.76	14

ตารางที่ ข.10 ผลการค้นหาคำตอบการปรับค่า PM กับ TSP ด้วย MXUPG

Algorithm	PM 0.01		PM 0.03		PM 0.05		PM 0.07		PM 0.09	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR	MBF	SR
BURMA14	3323	100	3323	100	3323	100	3323.13	99	3323	100
ULYSSES16	6861.87	78	6862.63	72	6861.71	79	6861.06	84	6861.83	77
GR17	2085.7	86	2085.65	87	2085.58	88	2085.25	95	2085.8	84
GR21	2714.67	92	2712.24	96	2707	100	2711.79	97	2708.5	98
ULYSSES22	7057.61	48	7059.1	46	7031.68	78	7054.3	51	7058.83	45
FRI26	955.49	27	955.99	23	952.85	25	956.09	27	955.99	27
BAYG29	1646.35	17	1650.28	20	1639.13	32	1651.74	18	1651.73	17
BAYS29	2051.83	13	2042.96	16	2040	18	2041.77	15	2051	14
ผลรวม		461		460		520		486		462

ตารางที่ ข.11 ผลการค้นหาคำตอบการปรับค่า TR กับ TSP ด้วย RPG

Algorithm	TR 50		TR 100		TR 500		TR 1000	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
BURMA14	3323.91	93	3323	100	3325.95	80	3325.34	82
ULYSSES16	6874.2	45	6863.02	68	6878.56	36	6882.34	34
GR17	2090.05	71	2086.48	83	2094.42	73	2097.81	62
GR21	2792.49	59	2726.27	79	2836.29	42	2824.46	43
ULYSSES22	7090.95	17	7077.65	25	7098.33	20	7110.21	9
FRI26	962.99	6	963.71	10	974.18	2	967.36	4
BAYG29	1696.09	0	1701.3	0	1738	0	1739.93	1
BAYS29	2072.34	0	2092.55	1	2088.2	1	2095.98	2
ผลรวม		291		366		254		237

ตารางที่ ข.12 ผลการค้นหาคำตอบการปรับค่า PR กับ TSP ด้วย RPG

Algorithm	PR 0.10		PR 0.30		PR 0.60		PR 0.90	
	MBF	SR	MBF	SR	MBF	SR	MBF	SR
BURMA14	3325.47	81	3325.47	81	3323	100	3323	100
ULYSSES16	6883.67	34	6879.77	34	6863.02	68	6880.82	26
GR17	2094.83	73	2094.55	73	2086.48	83	2090.33	79
GR21	2784.59	55	2801.06	54	2726.27	79	2727.22	77
ULYSSES22	7106.72	12	7106.08	17	7077.65	25	7156.42	10
FRI26	970.09	3	967.61	4	963.71	10	969.75	9
BAYG29	1736.59	0	1749.14	3	1701.3	0	1681.4	1
BAYS29	2090.06	1	2091.39	1	2092.55	1	2075.76	2
		259		267		366		304

จากตาราง ข.9 การกำหนดค่า RM ของ MXUPG เท่ากับ 20 และมีจำนวนอนุภาค 20 จะได้รับคำตอบที่ดีที่สุด โดยการกำหนดค่า RM เท่ากับ 40 และมีจำนวนอนุภาค 10 ได้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า RM เท่ากับ 20 และมีจำนวนอนุภาค 20 เพราะการกำหนด RM มากเกินไป อาจจะทำให้เกิดปัญหาการลู่เร็วเกินไป ทำให้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า RM ให้เหมาะสมกับการแก้ปัญหา โดยการกำหนดค่า RM เท่ากับ 10 และมีจำนวนอนุภาค 40 ได้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า RM เท่ากับ 20 และมีจำนวนอนุภาค 20 เพราะการกำหนด RM น้อยเกินไปอาจทำให้เกิดปัญหาการลู่ช้าเกินไป ทำให้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า RM ให้เหมาะสมกับการแก้ปัญหา ดังนั้นวิธานิพนธ์ฉบับนี้ได้เลือกใช้ค่า RM ของ MXUPG เท่ากับ 20 และมีจำนวนอนุภาค 20

จากตาราง ข.10 การกำหนดค่า PM ของ MXUPG เท่ากับ 0.05 จะได้รับคำตอบที่ดีที่สุด โดยการกำหนดค่า PM ให้มีค่าน้อยเกินไป ($PM = 0.01$ และ 0.03) อาจจะทำให้เกิดปัญหาการลู่ช้าเกินไปทำให้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า PM ให้เหมาะสม ในทางกลับกันการกำหนดค่า PM ให้มีค่ามากเกินไป ($PM = 0.07$ และ 0.09) อาจจะทำให้เกิดปัญหาการลู่ช้าเกินไปที่ขัดขวางกระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาคำตอบแย่กว่าการกำหนดค่า PM ให้เหมาะสม ดังนั้นวิธานิพนธ์ฉบับนี้ได้เลือกใช้ค่า PM เท่ากับ 0.05

จากตาราง ซ.11 การกำหนดค่า TR ของ RPG เท่ากับ 100 จะได้รับคำตอบที่ดีที่สุด โดยการกำหนดค่า TR ให้มีค่าน้อยเกินไป ($TR = 50$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาแย่งกว่าการกำหนดค่า TR ให้เหมาะสม ในทางกลับกันการกำหนดค่า TR ให้มีค่ามากเกินไป ($TR = 500$ และ 1000) อาจจะทำให้เกิดปัญหาการกลายพันธุ์น้อยเกินไปทำให้ผลลัพธ์การค้นหาแย่งกว่าการกำหนดค่า TR ให้เหมาะสม ดังนั้นวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า TR เท่ากับ 100

จากตาราง ซ.12 การกำหนดค่า PR ของ RPG เท่ากับ 0.6 จะได้รับคำตอบที่ดีที่สุด โดยการกำหนดค่า PR ให้มีค่าน้อยเกินไป ($PR = 0.1$ และ 0.3) อาจจะทำให้เกิดปัญหาการกลายพันธุ์น้อยเกินไปทำให้ผลลัพธ์การค้นหาแย่งกว่าการกำหนดค่า PR ให้เหมาะสม ในทางกลับกันการกำหนดค่า PR ให้มีค่ามากเกินไป ($PR = 0.9$) อาจจะทำให้เกิดปัญหาการกลายพันธุ์ที่ขัดขวางกระบวนการหาคำตอบทำให้ผลลัพธ์การค้นหาแย่งกว่าการกำหนดค่า PR ให้เหมาะสม ดังนั้นวิทยานิพนธ์ฉบับนี้ได้เลือกใช้ค่า PR เท่ากับ 0.6

ภาคผนวก ฅ

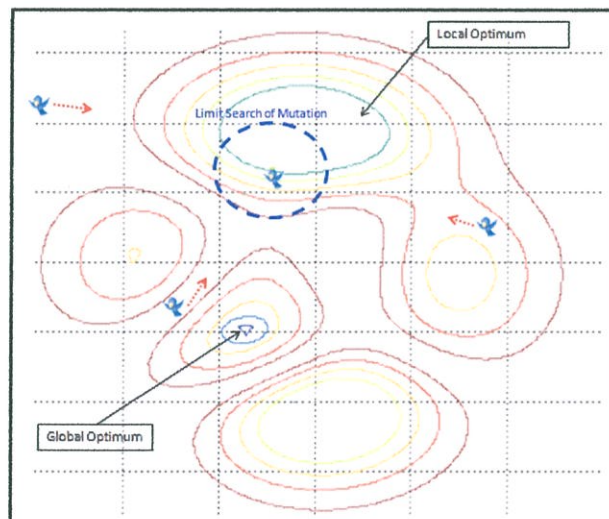
วิธีการกลายพันธุ์และรีโพอิชันแบบอื่นที่นำมาเปรียบเทียบกับวิธีการที่นำเสนอ

ในภาคผนวกนี้จะกล่าวถึงขั้นตอนวิธีอื่นที่นำมาเปรียบเทียบกับวิธีการที่นำเสนอ (MXUPG, RPG และ MRPSO) เพื่อเป็นการแสดงให้เห็นว่าวิธีการที่นำเสนอ นั้น สามารถเพิ่มประสิทธิภาพการค้นหาค้นหาของ PSO ได้อย่างมีประสิทธิภาพมากที่สุดเมื่อเปรียบเทียบกับวิธีการเพิ่มประสิทธิภาพวิธีอื่นที่มีขั้นตอนการดำเนินการใกล้เคียงกัน

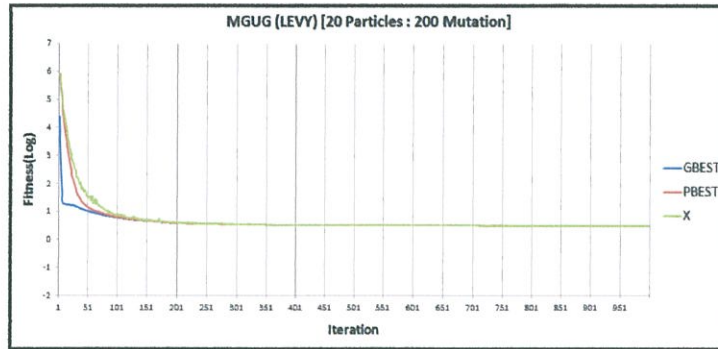
ฅ.1 วิธีการกลายพันธุ์แบบอื่นที่นำมาเปรียบเทียบกับวิธีการกลายพันธุ์ที่นำเสนอ

สำหรับวิธีการกลายพันธุ์ที่นำเสนอ นั้นคือ MXUPG ในบทที่ 4 นั้น MXUPG ได้ถูกเปรียบเทียบกับ MXUX คือ วิธีประยุกต์การกลายพันธุ์ประเภท CM โดยดำเนินการกลายพันธุ์รอบตำแหน่งของอนุภาค แล้วนำผลการกลายพันธุ์มาปรับปรุงตำแหน่งของอนุภาค PBEST และ GBEST แต่การกลายพันธุ์ประเภท CM นอกจากดำเนินการแบบ MXUX แล้วนั้น วิธีประยุกต์การกลายพันธุ์ประเภท CM สามารถดำเนินการกลายพันธุ์รอบ PBEST แล้วนำผลการกลายพันธุ์มาปรับปรุง PBEST และ GBEST หรือเรียกว่า MPUP และอีกในกรณีคือวิธีประยุกต์การกลายพันธุ์ประเภท CM ดำเนินการกลายพันธุ์รอบ GBEST แล้วนำผลการกลายพันธุ์มาปรับปรุง GBEST หรือเรียกว่า MGUG

ในกรณี CM ดำเนินการกลายพันธุ์กับ GBEST และผลการกลายพันธุ์มาปรับปรุง GBEST (PSO with mutation GBEST and update GBEST หรือ MGUG) วิธีนี้การกลายพันธุ์จะค้นหารอบจุดเพียงจุดเดียว คือ GBEST จึงมีจุดที่กระจายตัวค้นหาน้อย ดังรูป ฅ.1

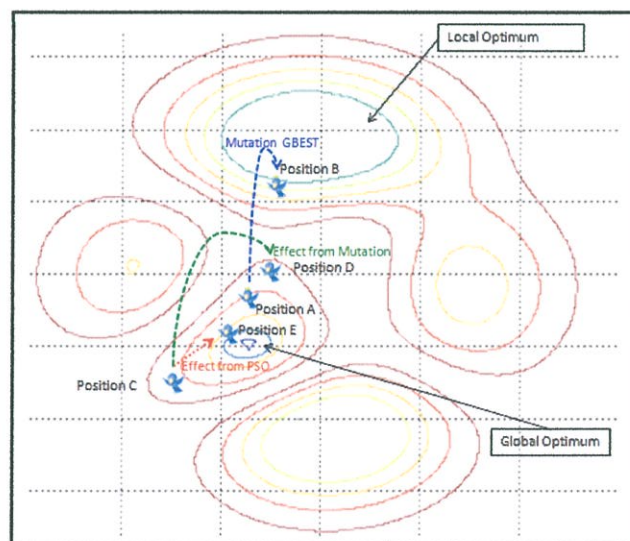


รูปที่ ฅ.1 วิธีการกลายพันธุ์ของ MGUG



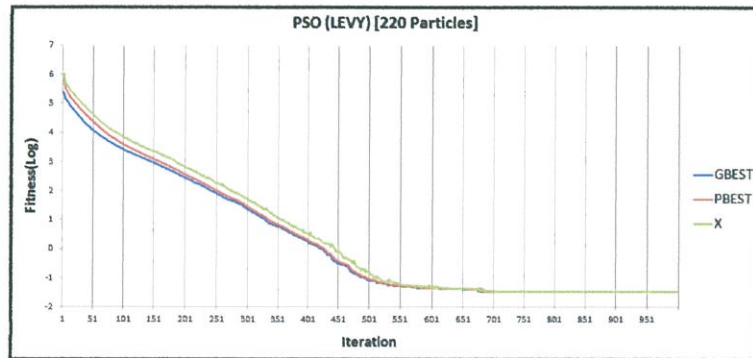
รูปที่ ๓.2 การลู่เข้าหากันของ MGUG

จากกราฟ ๓.2 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แสดงให้เห็นว่า เส้นกราฟของ GBEST ลดลงอย่างรวดเร็ว ส่งผลให้เส้นกราฟของ PBEST และ X ลดลงอย่างรวดเร็วตามแล้วเข้าสู่สถานะคงที่ โดยสาเหตุเนื่องจาก GBEST ถูกดำเนินการกลายพันธุ์เพียงตำแหน่งเดียวแต่มีจำนวนมาก ซึ่งทำให้ GBEST เคลื่อนที่แบบละโมบเป็นจำนวนมาก โดยจากกราฟ GBEST จะถูกดำเนินการกลายพันธุ์จำนวน 200 รอบ และมีจำนวนอนุภาค 20 อนุภาค ซึ่งเทียบเท่ากับ PSO จำนวน 220 อนุภาค ดังนั้นในหนึ่งรอบการค้นหาคำตอบ GBEST จะถูกปรับปรุงโดยการค้นหาแบบละโมบมากที่สุด 200 ครั้ง และถูกการปรับปรุงโดยการค้นหาแบบ PSO มากสุดเพียงหนึ่งครั้ง ส่วนอนุภาคทั้งหมดจะเคลื่อนที่ตาม GBEST เพียงแค่หนึ่งครั้ง จึงทำให้ GBEST มีค่าที่ดีที่สุดอย่างรวดเร็ว ซึ่ง GBEST จะเคลื่อนที่ออกจาก PBEST และ X เป็นระยะไกล โดยสมการการคำนวณความเร็วของ PSO คือ สมการ 2.2 และ 2.3 แสดงให้เห็นว่ายิ่งระยะทางระหว่าง GBEST กับ X มีระยะห่างกันมาก จะทำให้ผลต่างของ X กับ GBEST มีค่ามาก ส่งผลให้ความเร็วมีค่ามากอนุภาคจึงเคลื่อนที่เข้าหา GBEST เป็นระยะทางที่มากในการเคลื่อนที่หนึ่งครั้ง เสมือนกับอนุภาคกระโดดเข้าหา GBEST ดังรูป ๓.3 ซึ่งอนุภาคจะสูญเสียโอกาสในการค้นหาระหว่างทางที่ลู่เข้าหา GBEST เนื่องจากโดยปกติแล้วอนุภาคจะค่อยๆ เคลื่อนที่เข้าหา GBEST อย่างช้าๆ แล้วค้นหาคำตอบในระหว่างทางที่มันเคลื่อนที่



รูปที่ ๓.3 วิธีการปรับปรุงตำแหน่งอนุภาคของ MGUG

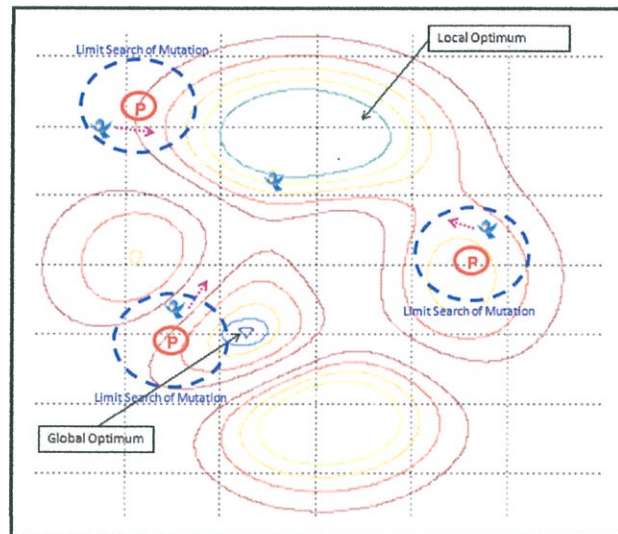
จากรูป ฅ.1 การกลายพันธุ์จะดำเนินการรอบ GBEST เพียงตำแหน่งเดียวเท่านั้น โดยมีขอบเขตภายในวงกลมสีน้ำเงิน และรูป ฅ.3 ผลจากการกลายพันธุ์ทำให้ GBEST เคลื่อนที่จากตำแหน่ง A ไปยังตำแหน่ง B เมื่อ GBEST ไปอยู่ตำแหน่ง B ซึ่งอยู่ไกลจากอนุภาคมาก ส่งผลให้อนุภาคกระโดดเข้าหา GBEST อนุภาคจึงเปลี่ยนจากตำแหน่ง C ไปยังตำแหน่ง D แล้วข้ามจุดสุดสัมบูรณ์ไป การค้นหาจึงไม่พบจุดสุดสัมบูรณ์ เนื่องจากอนุภาคเสียโอกาสในการค้นหาระหว่างทางที่ลู่ออกเข้าหา GBEST ในขณะที่ถ้าปล่อยให้ค้นหาโดยใช้ PSO ปกติ GBEST จะอยู่ในตำแหน่ง A อนุภาคอยู่ใกล้กับ GBEST จึงค่อยๆ เคลื่อนที่เข้าหา GBEST ไปยังตำแหน่ง E จึงทำให้พบจุดสุดสัมบูรณ์



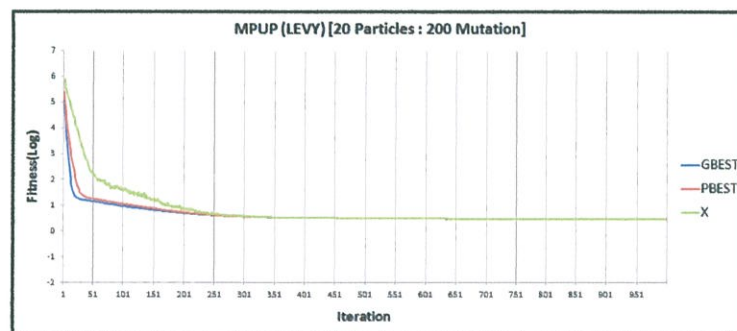
รูปที่ ฅ.4 การลู่ออกกันของ PSO

จากกราฟ ฅ.2 และ ฅ.4 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 เส้นกราฟทั้ง 3 เส้น (GBEST, PBEST, X) ของ MGUG เข้าสู่สภาพคงที่เร็วกว่าเส้นกราฟทั้ง 3 เส้น (GBEST, PBEST, X) ของ PSO นั้นหมายความว่า MGUG มีความเร็วในการลู่ออกกันมากกว่า PSO จึงประสบปัญหาการลู่ออกเร็วเกินไปรุนแรงกว่า PSO และผลจากที่ MGUG มีจุดที่กระจายตัวคั่นหาน้อย จึงทำให้บางปัญหา MGUG ได้ผลลัพธ์แยกว่า PSO โดยสังเกตได้จากตารางผลการทดลอง 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2

ในกรณี CM ดำเนินการกลายพันธุ์กับ PBEST และผลการกลายพันธุ์ใช้ปรับปรุง PBEST (PSO with mutation PBEST and update PBEST หรือ MPUP) วิธีนี้การกลายพันธุ์จะค้นหารอบ PBEST โดยการกลายพันธุ์มีหลายตำแหน่ง ดังรูป ฅ.5 การกระจายตัวจึงมากกว่า MGUG ดังนั้นวิธีนี้จึงมีประสิทธิภาพแก้ปัญหาหมกมอดได้ดีกว่า MGUG โดยสังเกตได้จากตารางผลการทดลอง 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2

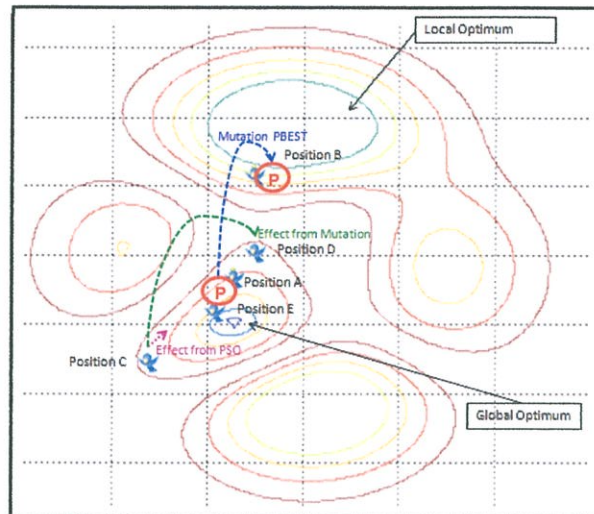


รูปที่ ๕.5 วิธีการกลายพันธุ์ของ MPUP



รูปที่ ๕.6 การคู่แข่งหาหกันของ MPUP

จากกราฟ ๕.6 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แสดงให้เห็นว่า เส้นกราฟของ GBEST และ PBEST ลดลงอย่างรวดเร็ว ส่งผลให้เส้นกราฟของ X ลดลงอย่างรวดเร็วตาม โดยสาเหตุเนื่องจาก PBEST เคลื่อนที่แบบละโมบและเมื่อ PBEST พบตำแหน่งที่ดีกว่า GBEST ก็จะทำ PBEST แทนค่า GBEST ซึ่งถึงว่า PBEST ได้ใช้วิธีการค้นหาแบบละโมบช่วยปรับปรุง GBEST ด้วย โดย PBEST ถูกดำเนินการกลายพันธุ์จำนวน 10 รอบ และมีจำนวนอนุภาค 20 อนุภาค ซึ่งเทียบเท่ากับ PSO จำนวน 220 อนุภาค ดังนั้นในหนึ่งรอบการค้นหาค่าตอบ PBEST จะถูกปรับปรุงโดยการค้นหาแบบละโมบมากที่สุด 10 ครั้ง และถูกปรับปรุงแบบ PSO มากสุดเพียงหนึ่งครั้ง ส่วนอนุภาคทั้งหมดจะเคลื่อนที่ตาม PBEST และ GBEST เพียงแค่หนึ่งครั้ง จึงทำให้ PBEST และ GBEST มีค่าที่ดีขึ้นอย่างรวดเร็ว ซึ่ง PBEST และ GBEST จะเคลื่อนที่ออกจาก X เป็นระยะที่มาก โดยสมการการคำนวณความเร็วของ PSO คือสมการ 2.2 และ 2.3 แสดงให้เห็นว่า ยิ่งระยะทางระหว่าง GBEST และ PBEST กับ X มีระยะห่างกันมาก จะทำให้ผลต่างของ X กับ GBEST และ PBEST มีค่ามาก ส่งผลให้ความเร็วมีค่ามากอนุภาคจึงเคลื่อนที่เข้าหา GBEST และ PBEST เป็นระยะทางที่มากในการเคลื่อนที่หนึ่งครั้ง เสมือนกับอนุภาคกระโดดเข้าหา GBEST และ PBEST ดังรูป ๕.7 ซึ่งอนุภาคจะสูญเสียโอกาสในการค้นหาระหว่างทางที่ลู่ออกเข้าหา GBEST และ PBEST



รูปที่ ๗.7 วิธีการปรับปรุงตำแหน่งอนุภาคของ MPUP

เมื่อเปรียบเทียบความเร็วในการลู่เข้าหากันของ PSO (กราฟ ๗.4) กับ MPUP (กราฟ ๗.6) เส้นกราฟทั้งสามเส้น (GBEST, PBEST, X) ของ MPUP เข้าสู่สภาพคงที่เร็วกว่าเส้นกราฟทั้งสามเส้น (GBEST, PBEST, X) ของ PSO นั้นหมายความว่า MPUP มีความเร็วการลู่เข้าหากันมากกว่า PSO เช่นเดียวกับ MGUG ดังนั้นจึงประสบปัญหาการลู่เร็วเกินไปมากกว่า PSO ปกติ จากเหตุผลเหล่านี้ทำให้บางปัญหา MPUP ได้ผลลัพธ์แย่กว่า PSO โดยสังเกตได้จากตารางผลการทดลองที่ 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2

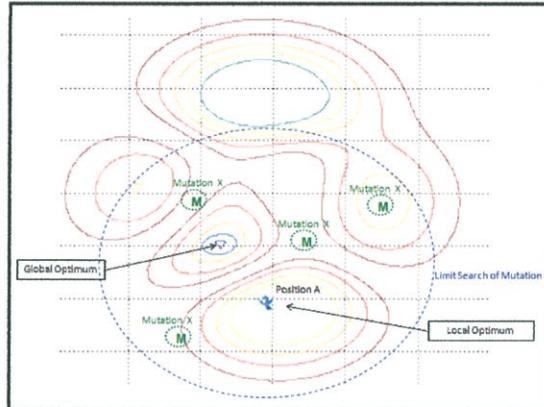
จากรูป ๗.5 การกลายพันธุ์แบบ MPUP การกลายพันธุ์จะดำเนินการรอบ PBEST โดย PBEST คือวงกลมสีแดงที่มีตัวอักษร P และขอบเขตของการกลายพันธุ์คือวงกลมสีน้ำเงิน การกลายพันธุ์มีการกระจายไปทั่วพื้นที่จึงมีโอกาสพบจุดสุดสัมบูรณ์มากกว่า MGUG

จากรูป ๗.7 ผลจากการกลายพันธุ์ทำให้ GBEST และ PBEST เคลื่อนที่จากตำแหน่ง A ไปยังตำแหน่ง B เมื่อ GBEST และ PBEST ไปอยู่ตำแหน่ง B ซึ่งอยู่ไกลจากอนุภาคมาก ส่งผลให้อนุภาคกระโดดเข้าหา GBEST และ PBEST อนุภาคจึงเปลี่ยนจากตำแหน่ง C ไปยังตำแหน่ง D แล้วข้ามจุดสุดสัมบูรณ์ไป การค้นหาจึงไม่พบจุดสุดสัมบูรณ์ เนื่องจากอนุภาคเสียโอกาสในการค้นหาระหว่างทางที่ลู่เข้าหา GBEST และ PBEST แต่ถ้าปล่อยให้ค้นหาโดยใช้ PSO ปกติ GBEST และ PBEST จะอยู่ในตำแหน่ง A อนุภาคอยู่ใกล้กับ GBEST และ PBEST จึงค่อยๆ เคลื่อนที่เข้าหา GBEST และ PBEST ไปยังตำแหน่ง E จึงพบจุดสุดสัมบูรณ์

ผลจากที่อนุภาคลู่เข้าหากันเร็วมากเกินไปจะส่งผลเสียกับปัญหามัลติโมดอล แต่สำหรับกรณีปัญหา uni-modal ผลจากการลู่เข้าหากันเร็วจะส่งผลดีกว่าเพราะพบคำตอบได้เร็วขึ้น จึงทำให้ CM ทั้ง 3 วิธี (MGUG, MPUP, MXUX) ได้ผลลัพธ์ที่ดีกว่า PSO โดยสังเกตได้จากตารางผลการทดลองที่ 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แต่ปัญหา uni-modal ไม่ใช่ปัญหาสำคัญของ PSO เพราะ PSO สามารถหาคำตอบพบเพียงแต่ใช้เวลาการค้นหามากกว่าวิธี CM ทั้งสาม แต่กรณีปัญหามัลติโมดอลในบางครั้งวิธี CM ทั้งสามจะก่อให้เกิดปัญหาการลู่เร็วเกินไปที่รุนแรงกว่า PSO ทำให้ติดจุดสุดสัมพัทธ์ได้ง่ายขึ้นจึงได้รับคำตอบแย่กว่า PSO ซึ่งปัญหาการลู่เร็วเกินไปนี้ไม่สามารถแก้ไขโดยเพิ่มเวลาการค้นหา

ฉ.2 การติดในจุดสุดสัมพัทธ์ของ PSO ที่ประยุกต์ใช้ CM

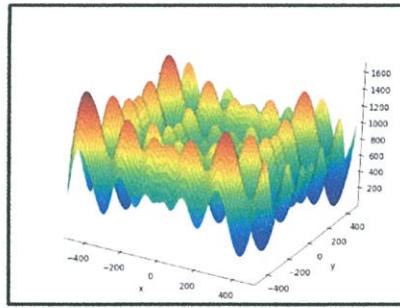
เนื่องจากคุณสมบัติของ MXUPG คือต้องการค้นหาลูกบอลจุดสุดสัมพัทธ์ให้มีขอบเขตที่กว้างกว่า PSO ปกติ แต่ขอบเขตการกลายพันธุ์ต้องมีขอบเขตที่ไม่กว้างมากนัก เพราะถ้าขอบเขตกว้างมากเกินไป การกลายพันธุ์จะสุ่มเป็นบริเวณกว้างเสมือนกับการค้นหาแบบสุ่ม ซึ่งส่วนมากแล้วผลที่ได้จากการค้นหาแบบสุ่มจะแยกว่าตำแหน่งจุดสุดสัมพัทธ์ที่อนุภาคติดอยู่ ทำให้ผลที่ได้จากการกลายพันธุ์ไม่สามารถนำมาใช้ได้การกลายพันธุ์จึงไม่มีประโยชน์ดังรูป ฉ.8



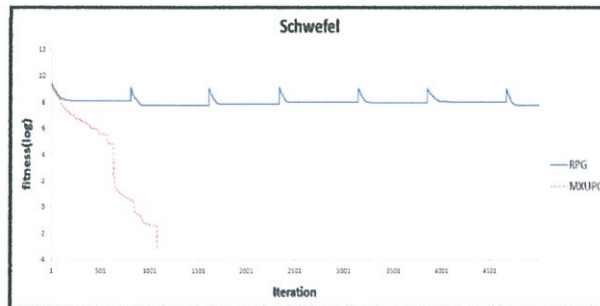
รูปที่ ฉ.8 สาธิตกำหนดขอบเขตการกลายพันธุ์มากเกินไป

จากรูป ฉ.8 ถ้ากำหนดขอบเขตการกลายพันธุ์มากเกินไป การกลายพันธุ์จะมีการสุ่มเป็นบริเวณกว้างจึงทำให้ผลจากการกลายพันธุ์ส่วนมากจะได้ตำแหน่งที่แยกว่าจุดสุดสัมพัทธ์ที่ติดอยู่

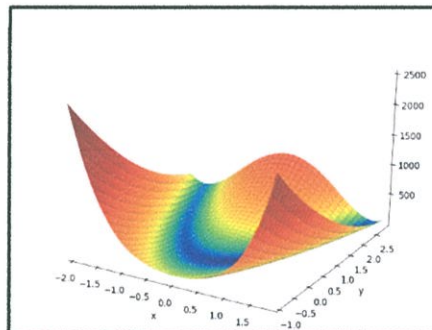
ปัญหาขอบเขตการกลายพันธุ์เกิดจากปัญหาที่ถูกนำมาแก้ไข ถ้ากรณีปัญหาที่ถูกนำมาแก้ไขมีจุดสุดสัมบูรณ์อยู่ใกล้จุดสุดสัมพัทธ์หรือขอบเขตของการกลายพันธุ์สามารถครอบคลุมจุดสุดสัมบูรณ์ ปัญหาขอบเขตการกลายพันธุ์จะไม่เกิดขึ้นกับ MXUPG ตัวอย่างเช่นฟังก์ชัน SCHWEFEL ในรูป ฉ.9 และจากกราฟ ฉ.10 ซึ่งมาจากผลการทดลองในหัวข้อ 5.7 พบว่าเส้นกราฟมีค่าลดลงแล้วพบจุดสุดสัมบูรณ์ แต่ถ้ากรณีปัญหาที่ถูกนำมาแก้ไขมีจุดสุดสัมบูรณ์อยู่ไกลจากจุดสุดสัมพัทธ์หรือขอบเขตของการกลายพันธุ์ไม่ครอบคลุมจุดสุดสัมบูรณ์ ปัญหาขอบเขตการกลายพันธุ์อาจจะเกิดขึ้นกับ MXUPG ตัวอย่างเช่นฟังก์ชัน ROSENBROCK ในรูป ฉ.11 และจากกราฟ ฉ.12 ซึ่งมาจากผลการทดลองในหัวข้อ 5.7 พบว่าเส้นกราฟลดลงและคงที่เหมือนกรณีปัญหาการติดในจุดสุดสัมพัทธ์ของ PSO นอกจากนี้จากตารางผลการทดลองที่ 5.9 ซึ่งมาจากผลการทดลองในหัวข้อ 5.4.2 แสดงให้เห็นว่า MXUPG ไม่สามารถหาคำตอบของปัญหาทั้งหมดในตารางพบ



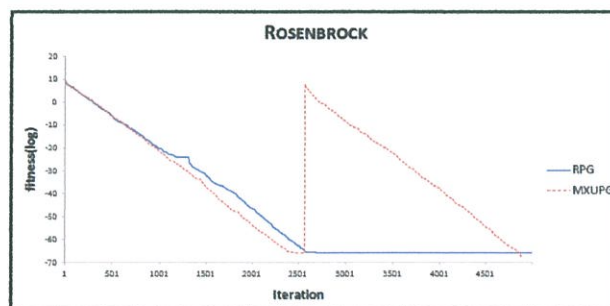
รูปที่ ๙.๙ ฟังก์ชัน SCHWEFEL [58]



รูปที่ ๙.๑๐ ผลการค้นหาของ MXUPG และ RPG ของ SCHWEFEL



รูปที่ ๙.๑๑ ฟังก์ชัน ROSENBROCK [58]

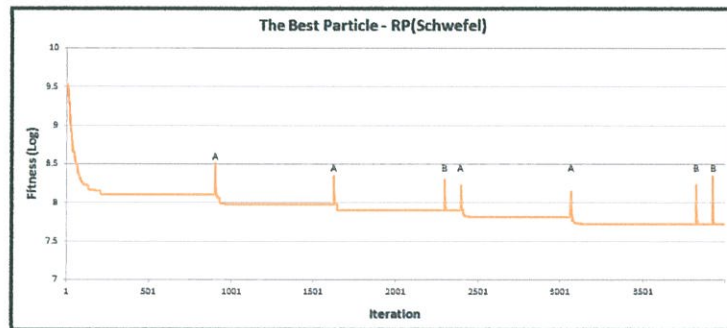


รูปที่ ๙.๑๒ ผลการค้นหาของ MXUPG และ RPG ของ ROSENBROCK

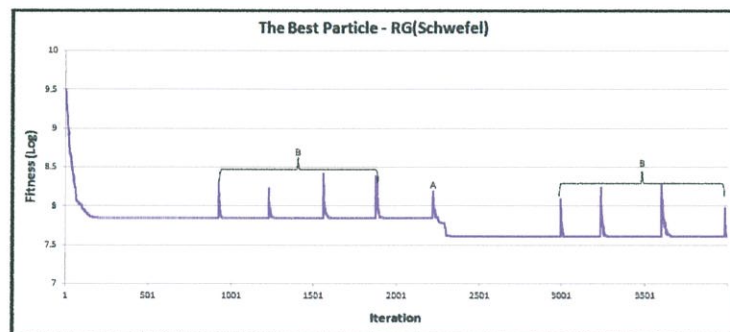
จากรูป ฅ.9 พบว่าฟังก์ชันนี้มีจุดสุดสัมพัทธ์กระจายอยู่ทั่วไปไปปริภูมิของคำตอบ ซึ่งตำแหน่งของจุดสุดสัมพัทธ์อยู่ใกล้เคียงกับจุดสุดสัมบูรณ์ ดังนั้นปัญหานี้ MXUPG จะสามารถแก้ปัญหาได้อย่างมีประสิทธิภาพ สำหรับรูป ฅ.11 พบว่าฟังก์ชันนี้มีจุดสุดสัมพัทธ์อยู่ที่ร่องด้านซ้ายและจุดสุดสัมบูรณ์อยู่ร่องด้านขวา ซึ่งตำแหน่งของจุดสุดสัมพัทธ์อยู่ห่างจากจุดสุดสัมบูรณ์มาก ดังนั้นปัญหานี้ MXUPG จะแก้ปัญหาได้ไม่ค่อยมีประสิทธิภาพ

ฅ.3 รีโพซิชันแบบอื่นที่นำมาเปรียบเทียบกับรีโพซิชันที่นำเสนอ

สำหรับวิธีรีโพซิชันที่นำเสนอ นั้นคือ RPG ในบทที่ 4 นั้น RPG ได้ถูกเปรียบเทียบกับ RN แต่รีโพซิชันยังมีวิธีการดำเนินการนอกจากแบบ RN ดังนั้นเพื่อแสดงให้เห็นผลกระทบจากการไม่รีเซ็ต PBEST และ GBEST วิทยานิพนธ์ฉบับนี้ จึงได้ทำการทดลอง โดยแบ่งกรณีทดลองออกเป็นดังนี้ รีโพซิชันที่รีเซ็ต GBEST แต่ไม่รีเซ็ต PBEST (Reposition with reset GBEST หรือ RG) รีโพซิชันที่รีเซ็ต PBEST แต่ไม่รีเซ็ต GBEST (Reposition with reset PBEST หรือ RP)



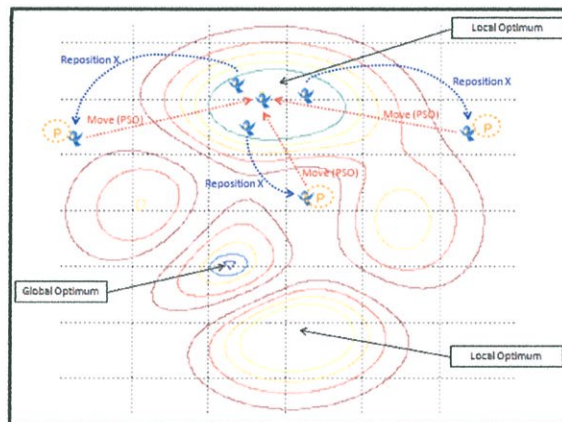
รูปที่ ฅ.13 ตัวอย่างของ RP



รูปที่ ฅ.14 ตัวอย่างของ RG

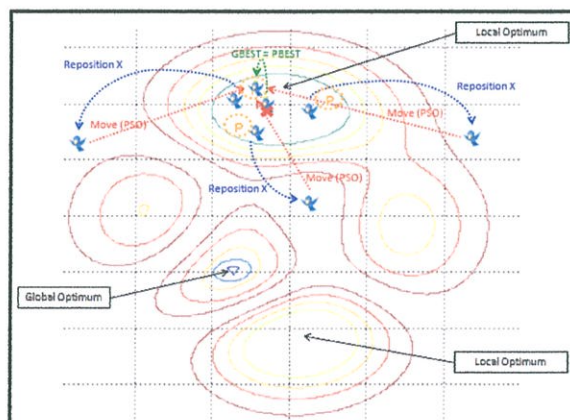
จากกราฟ ฅ.13 และ ฅ.14 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 เส้นกราฟในช่วงแรกจะมีค่าลดลงเนื่องจากการค้นหาด้วย PSO ทำให้ค่าความเหมาะสมดีขึ้น หลังจากนั้นเส้นกราฟจะเข้าสู่สภาพคงที่จนกระทั่งสามารถสรุปได้ว่าเวลานั้นได้เกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น หลังจากนั้นจะดำเนินการรีเซ็ตซึ่งจะทำให้เส้นกราฟสูงขึ้นเนื่องจากอนุภาคถูกกระจายตัวออกจากกันค่าความเหมาะสมของอนุภาคจึงแย่งลง หลังจากการรีเซ็ตจะค้นหาด้วย PSO ปกติจึงทำให้เส้นกราฟลดลง

จากกราฟ RP, RG อนุภาคจะเคลื่อนที่จากตำแหน่งที่มันกระจายตัวออกไปแล้วลู่อู่เข้าหากันมายังตำแหน่ง PBEST และ GBEST เดิมก่อนการรีเซ็ต ในระหว่างทางที่มันเคลื่อนที่กลับ ถ้าอนุภาคได้พบตำแหน่งที่ดีกว่า GBEST เดิม จะทำให้พบ GBEST ใหม่ จึงสามารถปรับปรุงคำตอบให้ดีขึ้นได้ ซึ่งสามารถสังเกตได้จากกรอบรีเซ็ตที่เริ่มต้นที่จุด A จากกราฟ ฅ.13 และ ฅ.14 และจากตารางผลการทดลองที่ 5.13 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 แสดงให้เห็นว่า RP, RG ได้รับคำตอบการค้นหาดีกว่า PSO ดังนั้นทั้ง 3 วิธี สามารถปรับปรุงคำตอบให้ดีและเพิ่มประสิทธิภาพการค้นหาของ PSO แต่โดยส่วนมากระหว่างทางที่อนุภาคจะลู่อู่เข้าหา GBEST เดิมจะไม่พบตำแหน่งที่ดีกว่า GBEST เดิม จึงไม่สามารถปรับปรุงคำตอบให้ดีขึ้น ซึ่งสามารถสังเกตได้จากกรอบรีเซ็ตที่เริ่มต้นที่จุด B จากกราฟ ฅ.13 และ ฅ.14 ที่เป็นเช่นนี้มาจากสาเหตุที่การค้นหาถูกบังคับให้ทิศทางลู่อู่กลับเข้าหา GBEST เดิม จึงทำให้การค้นหาหามีขอบเขตที่แคบอยู่ในบริเวณรอบ GBEST เดิมซึ่งเป็นการค้นหาในบริเวณเดิมซ้ำๆ จึงทำให้ผลลัพธ์ที่ได้ไม่ดีเท่าที่ควร ดังรูป ฅ.15 และ ฅ.16



รูปที่ ฅ.15 สาธิตของ RP

รูป ฅ.15 คือภาพสาธิตของ RP อนุภาคถูกดำเนินการรีโพซิชั่น ทำให้อนุภาคกระจายตัวออกจุดสุดสัมพัทธ์ตามลูกศรเส้นประสีน้ำเงิน และ PBEST คือวงกลมเส้นประสีส้มซึ่งถูกรีเซ็ตทำให้ PBEST กระจายตัวออกจากจุดสุดสัมพัทธ์มาอยู่ใกล้กับอนุภาคในตำแหน่งใหม่ สำหรับ GBEST คือ นกที่สวมมงกุฎอยู่ที่ตำแหน่งจุดสุดสัมพัทธ์เดิม เนื่องจาก GBEST ยังคงอยู่ในจุดสุดสัมพัทธ์เดิม เมื่อดำเนินการค้นหาด้วย PSO อนุภาคจะลู่อู่กลับเข้ามาจุดสุดสัมพัทธ์เดิมตามลูกศรเส้นประสีแดง



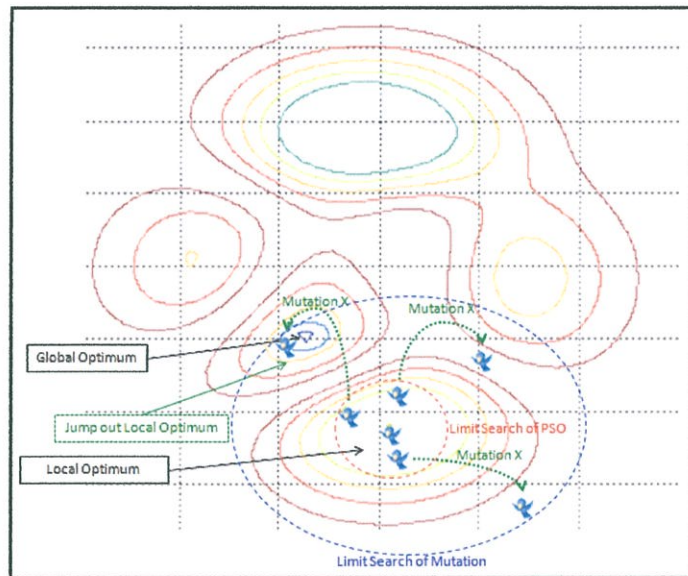
รูปที่ ฅ.16 สาธิตของ RG

รูป ฅ.16 คือภาพสาดิตของ RG อนุภาคถูกดำเนินการรีโพซิชัน ทำให้อนุภาคกระจายตัวออกจากจุดสุดสัมพัทธ์ตามลูกศรเส้นประสีน้ำเงิน สำหรับ PBEST คือวงกลมเส้นประสีส้ม และ GBEST คือ นกที่สวมมงกุฎซึ่งถูกรีเซ็ท แต่เนื่องจาก PBEST ไม่ถูกรีเซ็ท ตามหลักการทำงานของ PSO คือ PBEST ที่ดีที่สุดจะนำไปแทนที่ GBEST ทำให้ตำแหน่งของ GBEST เปลี่ยนไปตามลูกศรเส้นประสีเขียว GBEST ก็ยังคงอยู่ในจุดสุดสัมพัทธ์เดิม เมื่อดำเนินการค้นหาด้วย PSO อนุภาคจะถูกลบเข้ามาจุดสุดสัมพัทธ์เดิมตามลูกศรเส้นประสีแดง

ฅ.4 ปัญหาของรีโพซิชัน

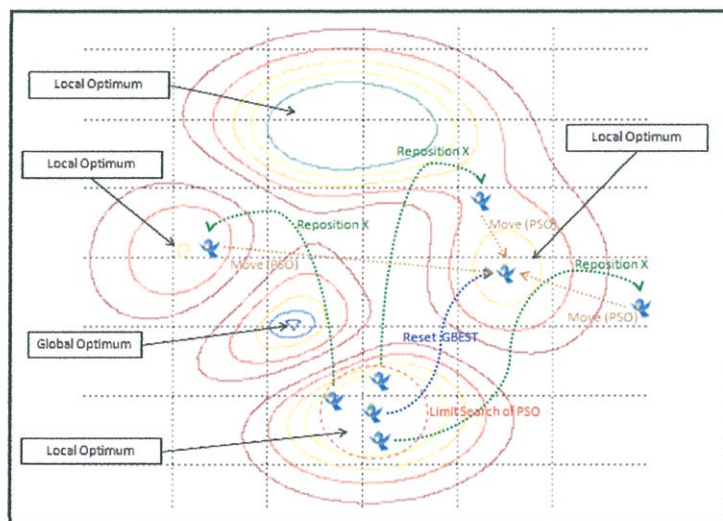
RPG สามารถปรับปรุงคำตอบของ PSO ได้โดยอาศัยหลักการ 2 อย่างดังนี้ หลักการแรกคือทำให้ PSO ได้มีโอกาสในการลองค้นหาคำตอบมากขึ้น โดยปกติ PSO เมื่อเกิดปัญหาการติดในจุดสุดสัมพัทธ์ขึ้น การค้นหาต่อไปก็จะมีประโยชน์ แต่ RPG สามารถทำการรีเซ็ทใหม่จึงทำให้ PSO ได้มีโอกาสค้นหาใหม่อีกครั้งจึงทำการค้นหาต่อไปนั้นมีประโยชน์ จึงทำให้ RPG มีโอกาสในการลองค้นหา มากกว่า PSO ซึ่งถ้าหาก PSO สามารถหาคำตอบพบเพียงครั้งเดียว RPG ก็จะสามารถหาคำตอบพบได้มากกว่าหรืออาจจะหาคำตอบพบทุกครั้ง จากตารางผลการทดลองที่ 5.13 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 แสดงให้เห็นว่าปัญหาที่ PSO สามารถหาคำตอบได้อย่างน้อยหนึ่งครั้ง RPG จะมีจำนวนที่พบคำตอบมากกว่า PSO เสมอ หลักการที่สอง คือผลจากการกลายพันธุ์ทำให้อนุภาคเคลื่อนที่ไปพบตำแหน่งใหม่ซึ่ง PSO ปกติไม่สามารถหาตำแหน่งเหล่านั้นพบ จึงทำให้ในบางกรณีที่ PSO หาคำตอบไม่พบแม้แต่ครั้งเดียว แต่ RPG สามารถหาคำตอบพบ จากตารางผลการทดลองที่ 5.13 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 แสดงให้เห็นว่าบางปัญหา PSO หาคำตอบไม่พบเลยแม้แต่ครั้งเดียวแต่ RPG สามารถหาคำตอบพบ

ถึงแม้ว่า RPG จะสามารถเพิ่มประสิทธิภาพการค้นหาคำตอบให้กับ PSO แต่บางปัญหา RPG ก็ไม่สามารถแก้ปัญหา โดยจากตารางผลการทดลองที่ 5.13 ซึ่งมาจากผลการทดลองในหัวข้อ 5.6 แสดงให้เห็นว่าบางปัญหา RPG ไม่สามารถหาคำตอบพบ เพราะปัญหาเหล่านี้ RPG แก้ปัญหาได้อย่างไม่มีประสิทธิภาพ เนื่องจาก RPG ไม่ได้เพิ่มความหลากหลายให้กับประชากรในขณะที่อนุภาคติดจุดสุดสัมพัทธ์ จึงทำให้ความหลากหลายของประชากรมีค่าเท่ากับ PSO ปกติซึ่งมีค่าน้อยมากทำให้อนุภาคค้นหาคำตอบเฉพาะในบริเวณจุดสุดสัมพัทธ์เท่านั้น ถ้ากรณีที่คำตอบอยู่ใกล้บริเวณจุดสุดสัมพัทธ์แล้วขณะนั้นอนุภาคมีความหลากหลายของประชากรมาก อนุภาคจะสามารถกระจายตัวออกไปค้นหาบริเวณรอบจุดสุดสัมพัทธ์ซึ่งอาจจะทำให้พบคำตอบดังรูป ฅ.17 แต่สำหรับ RPG อนุภาคจะติดอยู่ใน จุดสุดสัมพัทธ์เป็นเวลานานจนรีเซ็ทถูกดำเนินการทำให้อนุภาคกระจายตัวออกเป็นบริเวณกว้างมาก การค้นหาแทบที่จะค้นหาขอบบริเวณเดิมซึ่งเดิมอยู่ใกล้คำตอบแล้วก็กลับกลายเป็นไปค้นหาที่บริเวณอื่นที่ห่างไกลจากคำตอบแทนจึงไม่พบคำตอบ ดังรูป ฅ.18 ซึ่งเหตุการณ์นี้เรียกว่าปัญหาขาดความหลากหลาย (The lack Diversity Problem)



รูปที่ ๑๗.17 สาทิตกรณีประชากรมีความหลากหลายเมื่อติดจุดสุดสัมพัทธ์

จากรูป ๑๗.17 กรณีนี้อนุภาคติดอยู่ในจุดสุดสัมพัทธ์ ซึ่งจุดสุดสัมพัทธ์อยู่ใกล้กับจุดสุดสัมบูรณ์ โดยขอบเขตการค้นหาด้วย PSO จะอยู่ในวงกลมสีแดงเส้นประ จึงทำให้การค้นหาด้วย PSO ไม่สามารถหลุดออกจากจุดสุดสัมพัทธ์ แต่ถ้าดำเนินการกลายพันธุ์จะทำให้อนุภาคมีโอกาสกระจ่ายตัวออกจากจุดสุดสัมพัทธ์โดยขอบเขตการกลายพันธุ์อยู่ในวงกลมสีน้ำเงินเส้นประ ภายในขอบเขตของการกลายพันธุ์นี้ มีจุดสุดสัมบูรณ์อยู่จึงทำให้อนุภาคมีโอกาสที่จะสุ่มไปพบจุดสุดสัมบูรณ์และหลุดออกจากจุดสุดสัมพัทธ์เดิม



รูปที่ ๑๗.18 สาทิตของปัญหาขาดความหลากหลาย

จากรูป ฅ.18 กรณีนี้อุณหภูมิติดอยู่ในจุดสูงสุดสัมพัทธ์ ซึ่งจุดสูงสุดสัมพัทธ์อยู่ใกล้กับจุดสูงสุดสัมบูรณ์ โดยขอบเขตการค้นหาค้นหาด้วย PSO จะอยู่ในวงกลมสีแดงเส้นประ จึงทำให้การค้นหาค้นหาด้วย PSO ไม่สามารถหลุดออกจากจุดสูงสุดสัมพัทธ์ เมื่อดำเนินการ RPG อุณหภูมิกระจายตัวออกจากจุดสูงสุดสัมพัทธ์ เป็นบริเวณกว้างดังรูป แล้วลู่เข้าหา GBEST ในตำแหน่งใหม่ซึ่งจากรูปหลังจากการรีเซ็ต อุณหภูมิจะเคลื่อนที่ลู่เข้าหา GBEST ใหม่ ตามทิศทางลูกศรประสีน้ำตาล ผลจาก RPG ทำให้อุณหภูมิเคลื่อนที่ ออกจากบริเวณที่ใกล้กับจุดสูงสุดสัมบูรณ์และไปค้นหาคำตอบในจุดสูงสุดสัมพัทธ์อื่นที่ไกลจากจุดสูงสุดสัมบูรณ์ จากตัวอย่างนี้แสดงให้เห็นว่าในบางกรณี RPG แก้ปัญหาได้ไม่ค่อยมีประสิทธิภาพ

ปัญหาขาดความหลากหลายเกิดจากสาเหตุปัญหาที่นำมาแก้ไข ถ้าปัญหาที่นำมาแก้ไขมีจุดสูงสุดสัมบูรณ์อยู่ใกล้จุดสูงสุดสัมพัทธ์ ในกรณีนี้ RPG ทำให้อุณหภูมิกระจายตัวออกเป็นบริเวณกว้าง จึงไม่ค่อยมีประสิทธิภาพสำหรับการแก้ปัญหาเหล่านี้ดังรูป ฅ.9 จากกราฟ ฅ.10 พบว่าการค้นหาค้นหาด้วย RPG นั้นไม่ว่าจะรีเซ็ตกี่รอบก็ไม่ทำให้พบคำตอบถึงแม้ว่าระดับเส้นกราฟที่ติดจุดสูงสุดสัมพัทธ์จะมีการเปลี่ยนแปลงก็ตาม

แต่ถ้ากรณีปัญหาที่นำมาแก้ไขมีจุดสูงสุดสัมบูรณ์อยู่ไกลจากจุดสูงสุดสัมพัทธ์ RPG สามารถแก้ปัญหาเหล่านี้ได้อย่างมีประสิทธิภาพ เพราะ RPG จะทำให้อุณหภูมิกระจายตัวออกเป็นบริเวณกว้าง แล้วลู่เข้าไปค้นหาในบริเวณใหม่ จึงมีโอกาที่จะกระโดดออกจากจุดสูงสุดสัมพัทธ์เดิมแล้วมาค้นหาใน บริเวณพื้นที่ใหม่ซึ่งอาจจะพบจุดสูงสุดสัมบูรณ์ดังรูป ฅ.11 และจากกราฟ ฅ.12 พบว่าในรอบแรกของการค้นหาอุณหภูมิเกิดการติดจุดสูงสุดสัมพัทธ์ เมื่อทำการรีเซ็ตอุณหภูมิเกิดการกระจายตัวออกแล้วลู่กลับเข้าหากันใหม่ โดยไปค้นหาในบริเวณซึ่งมีจุดสูงสุดสัมบูรณ์อยู่จึงพบคำตอบ

ประวัติผู้เขียน

ชื่อ - นามสกุล	นายเฉียบวุฒิ รัตนวิไลสกุล
วัน เดือน ปีเกิด	10 มีนาคม 2531
การศึกษา	ปีการศึกษา 2543 - 2549 ระดับมัธยมศึกษา โรงเรียนหอวัง ปีการศึกษา 2550 - 2554 ระดับอุดมศึกษา คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2554 - 2556 ระดับบัณฑิตศึกษา คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง