



ใบรับรองวิทยานิพนธ์
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

..... วิทยาศาสตร์มหาบัณฑิต (วิทยาการคอมพิวเตอร์)

ปริญญา

..... วิทยาการคอมพิวเตอร์

..... วิทยาการคอมพิวเตอร์

สาขา

ภาควิชา

เรื่อง ขั้นตอนวิธีเชิงพันธุกรรมแบบบิโอดที่เข้ารหัสด้วยอัลกอริทึม LZW แบบผสม
Hybrid LZWGA

นามผู้วิจัย นางสาวสาวิตรี ฉ่ำมัย

ได้พิจารณาเห็นชอบโดย
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์วรเศรษฐ์ สุวรรณิก, วศ.ค.)

หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์ศิริกร จันทร์นวล, M.Sc.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

(รองศาสตราจารย์กัญญา ชีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ **เดือน** **พ.ศ.**

สืบสีงธี มทาวิตยาสัยเกษตรศาสตร์

วิทยานิพนธ์

เรื่อง

ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม LZW แบบผสม

Hybrid LZWGA

โดย

นางสาวสาวิตรี น้าลมัย

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

พ.ศ. 2553

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สาวิตรี น้าลมัย 2553: ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม
LZW แบบผสม ปรินญาวิทยาศาสตร์มหาบัณฑิต (วิทยาการคอมพิวเตอร์) สาขา
วิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก:
ผู้ช่วยศาสตราจารย์วรัศนี สุวรรณิก, วศ.ด. 63 หน้า

อัลกอริทึม LZWGA สามารถแก้ปัญหา OneMax และ RandomRoad ได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ประสิทธิภาพ อย่างไรก็ตาม คำตอบของปัญหาดังกล่าวมีแบบแผนสูง ดังนั้น ผู้วิจัยจึงนำเสนอ ปัญหาใหม่สองปัญหา โดยปัญหาใหม่นี้มีพื้นฐานมาจากปัญหาที่ใช้ทดสอบแบบเดิม แต่จะมีการ สุ่มคำตอบเข้ามาเพื่อให้ได้คำตอบที่ไม่มีแบบแผน ได้แก่ ปัญหา RandomMax และ RandomRoyalRoad จากผลการทดลองที่ได้ แสดงให้เห็นว่าการใช้ LZWGA เพียงอย่างเดียวไม่สามารถแก้ปัญหาแบบสุ่มได้

ดังนั้นผู้วิจัยจึงปรับปรุง LZWGA เพื่อทำให้ประสิทธิภาพในการแก้ปัญหาแบบสุ่มดีขึ้น โดยรวมการค้นหาเฉพาะที่ (local search) เข้าไป เรียกว่า Hybrid LZWGA จากการทดลอง พบว่า อัลกอริทึมการค้นหาเฉพาะที่ส่งผลให้ประสิทธิภาพโดยรวมในการหาคำตอบของ LZWGA ดีขึ้น โดยดีขึ้นจากการใช้ LZWGA เพียงอย่างเดียว ประมาณ 38 เปอร์เซ็นต์ในปัญหา RandomMax 55 เปอร์เซ็นต์ ในปัญหา RandomRoyalRoad ที่ขนาดบล็อกละ 4 และ 47 เปอร์เซ็นต์ในปัญหา RandomRoyalRoad ที่ขนาดบล็อกละ 8

ลายมือชื่อนิติ

ลายมือชื่อประธานกรรมการ

Sawitree Chumlamai 2010: Hybrid LZWGA. Master of Science (Computer Science),
Major Field: Computer Science, Department of Computer Science. Thesis Advisor:
Assistant Professor Worasait Suwannik, Ph.D. 63 pages.

LZWGA can solve OneMax and RoyalRoad problems very efficiently. However, the answers of these problem have high regularity. Therefore, we proposed two new randomly problems: RandomMax and RandomRoyalRoad. These problems are based on the traditional problems but the answers are generated randomly. LZWGA could not solve these randomly problems.

Therefore, we combine a local search algorithm to LZWGA to improved its performance. We named the algorithm “Hybrid LZWGA”. The experiment shows that local search algorithm makes significant improvement to the performance of LZWGA. Hybrid LZWGA performs 38% better in RandomMax problem. It performs 55% and 47% better in RandomRoyalRoad problem with 4-bit and 8-bit block size.

Student’s signature

Thesis Advisor’s signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงได้ โดยได้รับคำแนะนำ ข้อคิดเห็น และตรวจทาน
แก้ไขข้อผิดพลาดต่าง ๆ จนทำให้สมบูรณ์ยิ่งขึ้นจาก ดร.วรเศรษฐ์ สุวรรณิก ประธานกรรมการที่
ปรึกษา ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ผู้วิจัยขอน้อมรำลึกถึงพระคุณของบิดามารดา ที่ส่งเสริมและสนับสนุนการศึกษาของผู้วิจัย
เสมอมา ขอขอบพระคุณครูบาอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้แก่ผู้วิจัย
ขอขอบคุณพี่ ๆ เพื่อน ๆ ที่เป็นกำลังใจและช่วยเหลือมาโดยตลอดไว้ ณ ที่นี้ด้วย

ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ ที่ได้สนับสนุนด้านอุปกรณ์ และสถานที่ตลอด
ระยะเวลาในการทำวิทยานิพนธ์

สาวิตรี น้าลมัย
เมษายน 2553

สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำอธิบายสัญลักษณ์และคำย่อ	(5)
คำนำ	1
วัตถุประสงค์	3
การตรวจเอกสาร	4
อุปกรณ์และวิธีการ	17
อุปกรณ์	17
วิธีการ	17
ผลและวิจารณ์	25
ผล	25
วิจารณ์	37
สรุปและข้อเสนอแนะ	40
สรุป	40
ข้อเสนอแนะ	41
เอกสารและสิ่งอ้างอิง	42
ภาคผนวก	44
ประวัติการศึกษา และการทำงาน	63

สารบัญตาราง

ตารางที่		หน้า
1	พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomMax ระหว่าง LZWGA และ GA	22
2	พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad ระหว่าง LZWGA และ GA	23
3	การทดสอบความแตกต่างของค่าเฉลี่ยกรณีในกลุ่มตัวอย่างทั้งสองไม่อิสระต่อกัน จาก T-Test	31

สารบัญภาพ

ภาพที่		หน้า
1	ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)	4
2	ตัวอย่างโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรม	4
3	การไขว้เปลี่ยนของขั้นตอนวิธีเชิงพันธุกรรม	6
4	การกลายพันธุ์ของขั้นตอนวิธีเชิงพันธุกรรม	6
5	ขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (LZWGA)	7
6	ตัวอย่างโครโมโซมของ LZWGA	8
7	รหัสเทียมการกลายโครโมโซมด้วยอัลกอริทึม LZW	8
8	การกลายโครโมโซมด้วยอัลกอริทึม LZW	9
9	รหัสเทียมของการค้นหาเฉพาะที่แบบ Hill Climbing	10
10	รหัสเทียมของ Memetic Algorithm	11
11	ตัวอย่างโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (Compressed Genetic Algorithm)	15
12	ขั้นตอนการทำงานของ Hybrid LZWGA	18
13	รหัสเทียมของ RandomMax กรณีที่ค่าความสุ่มเป็น 1	19
14	รหัสเทียมของ RandomRoyalRoad กรณีที่ค่าความสุ่มเป็น 1	21
15	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 0	25
16	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 1	26
17	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 0 และขนาดบล็อกเท่ากับ 4	26
18	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 1 และขนาดบล็อกเท่ากับ 4	27
19	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 0 และขนาดบล็อกเท่ากับ 8	27

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
20	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 1 และขนาดบล็อกละ 8	28
21	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดที่บิบอัดอยู่	29
22	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดที่บิบอัดอยู่ ที่ขนาดบล็อกละ 4	30
23	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดที่บิบอัดอยู่ ที่ขนาดบล็อกละ 8	30
24	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง Hybrid LZWGA 4 แบบ	33
25	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA 4 แบบ ที่ขนาดบล็อกละ 4	34
26	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA 4 แบบ ที่ขนาดบล็อกละ 8	34
27	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง Hybrid LZWGA และ Hybrid GA	35
28	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA และ Hybrid GA ที่ขนาดบล็อกละ 4	36
29	เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA และ Hybrid GA ที่ขนาดบล็อกละ 8	36

คำอธิบายสัญลักษณ์และคำย่อ

- GA = Genetic Algorithm
- LZW = Lempel-Ziv-Welch Algorithm
- LZWGA = LZW Encoding in Genetic Algorithm
- โครโมโซมที่ถูกบีบอัด = โครโมโซมที่เข้ารหัสและสามารถคลายได้ด้วยอัลกอริทึมการคลายของ LZW
- โครโมโซมที่ถูกคลาย = โครโมโซมที่ผ่านการคลายด้วยอัลกอริทึมการคลายของ LZW

ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม LZW แบบผสม

Hybrid LZWGA

คำนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithms – GA) เป็นวิธีการที่ใช้ในการค้นหาผลเฉลย (solutions) เพื่อให้ได้ค่าที่เหมาะสมที่สุด (optimum point) โดยเป็นวิธีการที่เลียนแบบการวิวัฒนาการของสิ่งมีชีวิต (Mitchell, 1998) ซึ่งถูกคิดค้นโดย John Holland โดยได้ประยุกต์กระบวนการเปลี่ยนแปลงของสิ่งมีชีวิตตามธรรมชาติมาใช้ร่วมกับการเขียนโปรแกรม เนื่องจากสิ่งมีชีวิตจะมีทั้งลักษณะที่ดีและไม่ดี แต่ลักษณะที่ดีจะได้รับการสนับสนุนให้มีการถ่ายทอดลักษณะทางพันธุกรรมไปยังรุ่นถัดไปเพื่อให้ได้สิ่งมีชีวิตรุ่นใหม่ที่ดีขึ้นกว่าเดิม โดยในการทำขั้นตอนวิธีเชิงพันธุกรรมนี้ จะต้องแปลงปัญหาให้อยู่ในรูปแบบของโครโมโซม ซึ่งโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมมีลักษณะเป็นสายบิต (binary string) เมื่อขนาดของโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหา (search space) ก็จะมีขนาดใหญ่ขึ้นตามไปด้วย

LZWGA ใช้วิธีการเข้ารหัสโครโมโซมแบบบีบอัด ทำให้โครโมโซมมีขนาดเล็กลง การวิวัฒนาการจะกระทำกับโครโมโซมที่ถูกบีบอัด ในขณะที่การวัดค่าความเหมาะสมของโครโมโซมจะกระทำกับโครโมโซมที่ถูกคลายด้วยอัลกอริทึมการคลายของ LZW (นริศ, 2548) โครโมโซมที่ถูกบีบจะอัดอยู่ในรูปของอาร์เรย์ของจำนวนเต็ม เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำโครโมโซมนี้มาคลายจะได้โครโมโซมของ GA ปกติคือ 1010101

LZWGA สามารถแก้ปัญหาขนาดใหญ่มากถึง 1 ล้านบิต (นริศ, 2548) ปัญหาดังกล่าวคือ OneMax, RoyalRoad และ Trap โดยคำตอบของปัญหาดังกล่าวจะเป็นคำตอบที่มีแบบแผนสูง คือ มีรูปแบบ (pattern) ที่ซ้ำกันสูง เช่น คำตอบเป็น 1 หมดทุกบิต ทำให้เกิดแนวคิดที่ว่า หากอัลกอริทึมมีประสิทธิภาพดีจริง ไม่ว่าจะนำไปทดสอบกับปัญหาใด ประสิทธิภาพย่อมไม่ลดลงไป ดังนั้นเพื่อเป็นการพิสูจน์ประสิทธิภาพของ LZWGA จึงไม่สามารถใช้ปัญหามาตรฐานที่นิยมใช้กันในการวิจัยด้าน GA ได้

ในปี พ.ศ.2550 ผู้วิจัยได้นำเสนอปัญหาใหม่ สำหรับทดสอบประสิทธิภาพของ LZWGA หรือขั้นตอนวิธีเชิงพันธุกรรมแบบบิพอดใดๆ โดยปัญหาที่นำเสนอมีพื้นฐานมาจากปัญหามาตรฐาน แต่ใช้การสุ่มคำตอบเพื่อให้ได้คำตอบที่ไม่มีแบบแผน โดยได้ตั้งสมมติฐานภายใต้ งานวิจัยของ LZWGA ที่สามารถทำงานได้เร็วกว่า GA ทั่วไปว่า ถ้า LZWGA สามารถทำงานได้เร็วกว่า GA แล้ว ดังนั้น ถ้านำ LZWGA ไปแก้ปัญหาก็ GA เคยแก้ได้ น่าจะทำให้การทำงานเร็วขึ้นกว่าเดิม แต่ผลการทดลองไม่เป็นไปตามสมมติฐานที่ตั้งไว้ ซึ่งจากผลการทดลองทำให้เห็นว่า LZWGA มีประสิทธิภาพในการทำงานดีต่อกว่า GA ในปัญหาแบบสุ่ม

ขั้นตอนวิธีเชิงพันธุกรรมถูกพัฒนาโดยนำไปทำงานร่วมกับเทคนิคการค้นหาเฉพาะที่แบบ hill-climbing เพื่อนำไปใช้ในการจัดการวางสอบของมหาวิทยาลัย ซึ่งนำชุดข้อมูลจริงมาจากมหาวิทยาลัยโฮจิมิน โดยผลที่ได้แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมผสมกับเทคนิคการค้นหาเฉพาะที่สามารถแก้ปัญหการจัดการวางสอบที่มีความยุ่งยากได้ (Nguyen and Duong, 2005)

วิทยานิพนธ์ฉบับนี้ จึงได้จัดทำขึ้นพัฒนาอัลกอริทึม LZWGA ให้สามารถแก้ปัญหาแบบสุ่มได้ โดยได้รวมอัลกอริทึมการค้นหาเฉพาะที่ (local search) เข้ามาเพื่อปรับปรุงค่าภายในโครโมโซมที่ได้อยู่แล้วให้ดียิ่งขึ้นและเพื่อเพิ่มประสิทธิภาพในการหาคำตอบ เรียกว่า Hybrid LZWGA

วัตถุประสงค์

1. เพื่อศึกษาวิธีขั้นตอนเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม LZW ที่สามารถเพิ่มประสิทธิภาพในการค้นหาคำตอบที่ดีที่สุดได้
2. เพื่อเปรียบเทียบและปรับปรุงประสิทธิภาพในการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม LZW เพื่อนำไปใช้ในการแก้ปัญหาแบบสุ่ม

ขอบเขตและข้อจำกัด

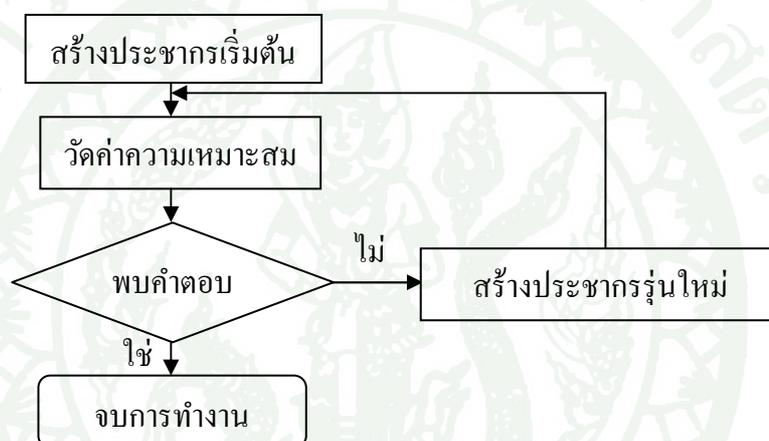
1. ใช้การค้นหาเฉพาะที่เพื่อทำงานร่วมกับ LZWGA ในการทดลองกับปัญหาแบบสุ่ม คือ RandomMax และ RandomRoyalRoad
2. ใช้จำนวนรุ่นและค่าความเหมาะสมในการประมวลผลในการวัดประสิทธิภาพ

การตรวจเอกสาร

ทฤษฎีที่เกี่ยวข้อง

1. ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm - GA)

ขั้นตอนวิธีเชิงพันธุกรรม เป็นวิธีการค้นหาที่เลียนแบบวิวัฒนาการของสิ่งมีชีวิต ถูกคิดค้นโดย Holland ในปี 1975 โดยมีขั้นตอนการทำงานดังภาพที่ 1



ภาพที่ 1 ขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)

การสร้างประชากรเริ่มต้น

ในการทำงานของขั้นตอนวิธีเชิงพันธุกรรม ต้องนำปัญหามาแปลงให้อยู่ในรูปแบบโครโมโซม โครโมโซมหลาย ๆ ตัวรวมเรียกว่า ประชากร โดยขั้นตอนวิธีเชิงพันธุกรรม สร้างประชากรเริ่มต้นโดยการสุ่ม ซึ่งประชากรจะมีลักษณะเป็นสายบิตของสตริง คือ มีค่า 0 และ 1 โดยเรียกสายบิตนี้ว่า โครโมโซม ดังภาพที่ 2

1	0	1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---

ภาพที่ 2 ตัวอย่างโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมการวัดค่าความเหมาะสม

การวัดค่าความเหมาะสม (fitness function) เป็นการให้คะแนนแก่โครโมโซมแต่ละตัว โดยจะต้องสอดคล้องกับวัตถุประสงค์ของปัญหา (objective function) ที่ได้กำหนดไว้ โครโมโซมแต่ละตัวจะมีค่าความเหมาะสมแตกต่างกันไป โครโมโซมที่มีค่าความเหมาะสมมาก จะมีโอกาสอยู่รอด และถูกนำไปใช้เพื่อสร้างประชากรในรุ่นถัดไป

ตัวอย่างการวัดค่าความเหมาะสม ในปัญหา OneMax การวัดค่าความเหมาะสมเกิดจากการนับจำนวนบิตที่มีค่าเป็น 1 ในโครโมโซม ดังนั้น ถ้าในโครโมโซม มีจำนวนเลข 1 มากเท่าใด โครโมโซมนั้นก็มีความเหมาะสมมากขึ้นเท่านั้น โดยค่าความเหมาะสมที่มากที่สุดต้องมีค่าไม่เกินความยาวของโครโมโซมนั้น

พบคำตอบ

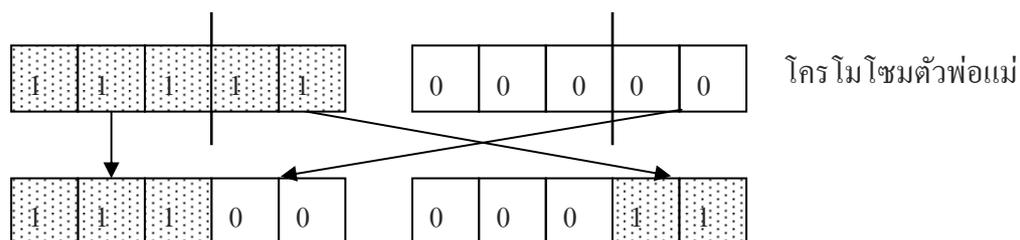
หลังจากนำโครโมโซมแต่ละตัวไปวัดค่าความเหมาะสมแล้ว ต้องมีการนำมาตรวจสอบเงื่อนไขในการหยุดการทำงาน ถ้าค่าความเหมาะสมที่ได้ มีค่าเท่ากับค่าความเหมาะสมที่กำหนดไว้ของปัญหานั้น ๆ ถือว่าพบคำตอบของปัญหา จะจบการทำงาน หรืออีกกรณีหนึ่งคือ ค่าความเหมาะสมไม่เป็นไปตามที่ตั้งไว้ แต่จำนวนรุ่นในการวิวัฒนาการเท่ากับจำนวนรุ่นที่กำหนดไว้ (max generation) จะจบการทำงานเช่นกัน

การสร้างประชากรรุ่นใหม่

ตัวดำเนินการที่นิยมนำไปใช้ในการสร้างประชากรรุ่นใหม่ของขั้นตอนวิธีเชิงพันธุกรรมหลัก ๆ ได้แก่ การไขว้เปลี่ยน (crossover) และ การกลายพันธุ์ (mutation)

การไขว้เปลี่ยน (Crossover)

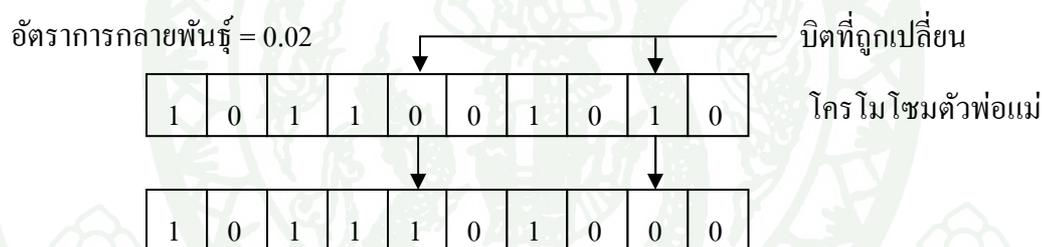
การไขว้เปลี่ยน เป็นกระบวนการที่สำคัญของขั้นตอนวิธีเชิงพันธุกรรม ในทางพันธุศาสตร์ เมื่อเกิดการไขว้เปลี่ยนจะทำให้เกิดการเปลี่ยนแปลงของสิ่งมีชีวิตที่หลากหลาย ทำให้ได้โครโมโซมรุ่นใหม่ขึ้นมาที่มีความแตกต่างไปจากโครโมโซมรุ่นก่อนหน้า



ภาพที่ 3 การไขว้เปลี่ยนของขั้นตอนวิธีเชิงพันธุกรรม

การกลายพันธุ์ (Mutation)

การกลายพันธุ์ เป็นการเปลี่ยนแปลงค่าบิตภายในโครโมโซมนั้น (เปลี่ยนจาก 0 เป็น 1 หรือ 1 เป็น 0) ทำให้โครโมโซมมีค่าเปลี่ยนแปลงไป โดยไม่ต้องอาศัยโครโมโซมตัวพ่อแม่ (parent chromosome)

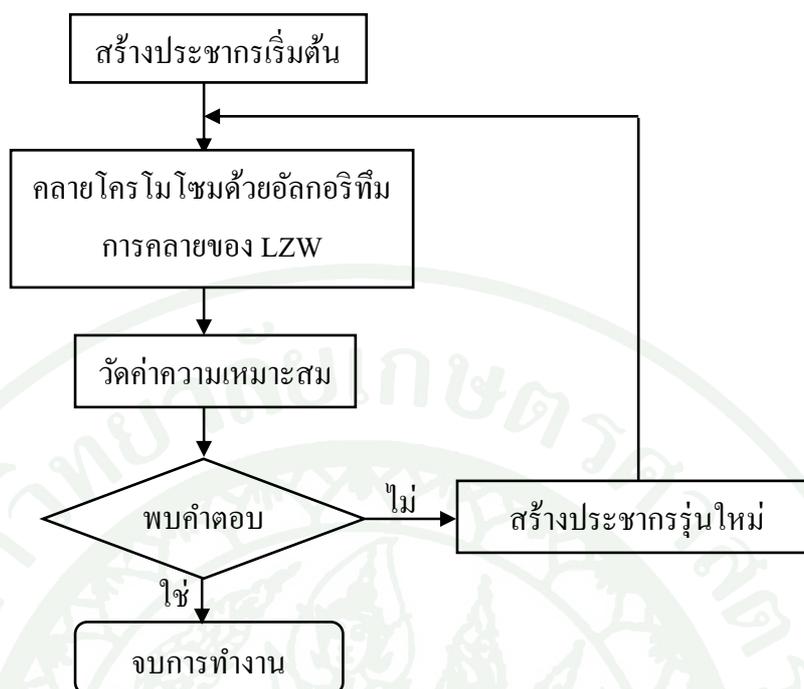


ภาพที่ 4 การกลายพันธุ์ของขั้นตอนวิธีเชิงพันธุกรรม

2. ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสแบบ LZW (LZWGA)

การเข้ารหัสแบบ LZW เป็นอัลกอริทึมที่พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม โดย LZWGA เข้ารหัสโครโมโซมในรูปแบบที่สามารถคลายได้ด้วยอัลกอริทึม Lempel-Ziv-Welch (LZW) ซึ่งช่วยลดขนาดของโครโมโซมและทำให้อัลกอริทึมสามารถแก้ปัญหามหาศาลได้

การทำงานของ LZWGA คล้ายกับการทำงานของ GA แต่ LZWGA จะมีกระบวนการคลายโครโมโซมเพิ่มเข้ามา โดยจะคลายโครโมโซมก่อนแล้วจึงวัดค่าความเหมาะสม ดังแสดงในภาพที่ 5



ภาพที่ 5 ขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (LZWGA)

สร้างประชากรเริ่มต้น

โครโมโซมในรุ่นแรกของ LZWGA สร้างขึ้นโดยการสุ่มโดยให้อยู่ในเงื่อนไขที่สามารถนำไปคลายได้เงื่อนไขดังกล่าวคือตำแหน่งที่ 1 ของโครโมโซมที่ถูกบีบอัดจะมีค่า 0 หรือ 1 ส่วนตำแหน่งถัดไปจะมีค่าได้ไม่เกินตำแหน่งนั้น เช่น โครโมโซมที่ถูกบีบตำแหน่งที่ 1 มีค่าอยู่ใน $\{0, 1\}$ โครโมโซมตำแหน่งที่ 2 มีค่าอยู่ใน $\{0, 1, 2\}$ ซึ่งสามารถสรุปได้ตามสมการที่ 1

$$a_i = r \bmod (i + 1), i \in \{1..n\} \quad (1)$$

โดย

a_i คือ ตำแหน่งที่ i ของโครโมโซมที่บีบอัดอยู่

r คือ เลขที่ถูกสุ่มขึ้นมา

n คือ ขนาดของโครโมโซม

0	1	0	2	2	3	4	1	5	8
---	---	---	---	---	---	---	---	---	---

ภาพที่ 6 ตัวอย่างโครโมโซมของ LZWGA

การคลายโครโมโซม และการวัดค่าความเหมาะสม

โครโมโซมเริ่มต้นที่เกิดจากการสุ่มของ LZWGA อยู่ในรูปแบบที่บีบอัดอยู่ ดังนั้นก่อนนำโครโมโซมนี้ไปวัดค่าความเหมาะสมจะต้องมีการคลายโครโมโซมก่อน ด้วยอัลกอริทึมการคลาย (นริศ, 2548) ดังภาพที่ 7 และวิธีการคลายดังภาพที่ 8 ซึ่งหลังจากผ่านอัลกอริทึมการคลายแล้วจะได้สายบิตเพื่อนำไปวัดค่าความเหมาะสมต่อไป

```

LZW-Decompress ()
  add entries 0,1 to the dictionary;
  read  $p$  and output one character corresponding to it;
  while  $c$  are still left
    if  $c$  is not in the dictionary
      enter in dictionary  $str(p) + fc(str(p))$ ;
      output  $str(p) + fc(str(p))$ ;
    else
      enter in dictionary  $str(p)+fc(str(c))$ ;
      output  $str(c)$ ;
   $p = c$ ;

```

ภาพที่ 7 รหัสเทียมการคลายโครโมโซมด้วยอัลกอริทึม LZW

โครโมโซม LZW

0	2	1	3	1	1
---	---	---	---	---	---

ตารางเก็บสายอักขระ

ขั้นตอนการทำงาน	p	c	ผลลัพธ์	ตารางเก็บสายอักขระ	
				ดัชนี	ค่าที่เก็บ
				0	0
				1	1
1	0	0	0	-	-
2	0	2	00	2	00
3	2	1	1	3	001
4	1	3	001	4	10
5	3	1	1	5	0011
6	1	1	1	6	11

โครโมโซมที่คลายแล้ว

0	0	0	1	0	0	1	1	1
---	---	---	---	---	---	---	---	---

ภาพที่ 8 การคลายโครโมโซมด้วยอัลกอริทึม LZW

การค้นหาเฉพาะที่

ขั้นตอนวิธีเชิงพันธุกรรมเป็นการค้นหาแบบครอบคลุม (global search) ซึ่งจะทำการค้นหาในปริภูมิการค้นหาโดยรวมได้ง่ายแต่เมื่อโครโมโซมมีขนาดใหญ่ขึ้น การค้นหาคำตอบจะใช้เวลานานมากขึ้น การค้นหาเฉพาะที่จะช่วยในการปรับปรุงคำตอบของประชากรที่ได้อยู่แล้ว โดยจะทำการวนซ้ำและเปลี่ยนค่าคำตอบของประชากรที่ได้อยู่แล้วให้ยิ่งดีขึ้นไปอีก เนื่องจากคำตอบที่ได้อยู่แล้วมักมีส่วนที่ไม่ดีรวมอยู่ด้วย

```

HillClimbing(current)
begin
    local variables: neighbor, a node
    current ← Make-Node(Initial-State[problem])
    loop do
        neighbor ← a highest-valued successor of current
        if value[neighbor] ≤ value[current]
            then return state[current]
        current ← neighbor
    end do
    return current
end

```

ภาพที่ 9 รหัสเทียมของการค้นหาเฉพาะที่แบบ Hill Climbing

การทำงาน

รับค่าตัวแปร *current* เข้ามา โดย *current* เป็นโครโมโซมที่เกิดจากการสุ่ม แล้วนำ *current* ไปวนซ้ำเปรียบเทียบกับโครโมโซมเพื่อนบ้าน (*neighbor*) แต่ละตัว โดย *neighbor* เกิดจากการนำ *current* ไปทำการกลายพันธุ์ทีละบิต กลายพันธุ์บิตแรกจะได้ *neighbor* ตัวที่หนึ่ง กลายพันธุ์บิตที่สองได้ *neighbor* ตัวที่สอง เช่นนี้ไปเรื่อย ๆ เท่ากับความยาวของโครโมโซมของ *current* เช่น ถ้า *current* ยาว 1000 บิต จะได้ *neighbor* 1000 ตัว เป็นต้น โดยถ้า *current* มีค่าความเหมาะสมมากกว่า *neighbor* จะคืนค่าความเหมาะสมของ *current* แต่ถ้า *current* มีค่าความเหมาะสมน้อยกว่า *neighbor* จะปรับค่า *current* ให้มีค่าเท่ากับ *neighbor* ทำเช่นนี้ไปเรื่อย ๆ จนกว่าจะเป็นไปตามเงื่อนไขการวนซ้ำ

Memetic Algorithm

ขั้นตอนวิธีนี้ได้แนวคิดมาจากทฤษฎี Memetics (Dawkins, 1976) และจากนั้นได้ถูกนำเสนอโดย (Moscato, 1989) เป็นการพัฒนาขั้นตอนวิธีเชิงพันธุกรรมให้ทำงานร่วมกับการค้นหาเฉพาะที่ โดยโครโมโซมในวิธีนี้มีติดนี้จะถูกเรียกว่า มีม (meme) (Dawkins, 1976) ในการนำเทคนิคการค้นหาเฉพาะที่เข้ามามีส่วนทำงานร่วมกับ GA เพื่อเพิ่มประสิทธิภาพในการหาคำตอบ โดยการทำงานอาจจัดกลุ่มเป็นกลุ่มย่อย แล้วใช้การค้นหาเฉพาะที่หาคำตอบที่ดีที่สุดในกลุ่ม (local optimum) จากนั้นใช้ GA เพื่อหาคำตอบที่ดีที่สุดของทั้งหมด (global maximum)

```

MemeticAlgorithm()
begin
    foundSolution ← false
    individual ← getIndividual(createFirstGeneration())
    generation ← 1
    HillClimbing(individual)
    while (true)
        evaluateFitness()
        if ((foundSolution=false) || (generation<MAXGENERATION))
            break;
        individual ← getIndividual(createNextGeneration())
        generation++;
        HillClimbing(individual)
    end while
    return individual
end

```

ภาพที่ 10 รหัสเทียมของ Memetic Algorithm

จากภาพที่ 10 มีการนำอัลกอริทึม HillClimbing มาทำงานร่วมกับขั้นตอนวิธีเชิงพันธุกรรม โดยขั้นตอนการทำงานจะทำขั้นตอนวิธีเชิงพันธุกรรมทั่วไปคือ สร้างประชากรรุ่นแรกโดยการสุ่มเก็บไว้ในตัวแปร individual หลังจากนั้นจะนำ individual แต่ละตัวส่งไปทำ HillClimbing เพื่อปรับปรุงค่าของประชากรแต่ละตัวให้ดีขึ้น แล้ววัดค่าความเหมาะสมและตรวจสอบว่าเป็นไปตามเงื่อนไขให้หยุดการทำงานหรือยัง ถ้ายังก็ทำการวนซ้ำสร้างประชากรรุ่นใหม่ แล้วนำประชากรแต่ละตัวไปไปทำ HillClimbing และวัดค่าความเหมาะสมเช่นนี้ไปเรื่อย ๆ จนเป็นไปตามเงื่อนไขให้หยุดการทำงาน ผลลัพธ์ที่ได้คือ ค่าประชากรตัวที่มีค่าความเหมาะสมมากที่สุด



งานวิจัยที่เกี่ยวข้อง

การหาคำตอบที่เหมาะสมที่สุดได้มีการนำกระบวนการทางพันธุศาสตร์เข้ามาใช้ในกระบวนการหาคำตอบ หรือที่เรียกว่า ขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)

ขั้นตอนวิธีเชิงพันธุกรรมเป็นเทคนิคหนึ่งในการค้นหาข้อมูลที่ได้รับความสนใจและให้ความแม่นยำในการค้นหา ซึ่งถูกคิดค้นโดย John Holland ขั้นตอนวิธีเชิงพันธุกรรม เป็นวิธีการค้นหาคำตอบที่เหมาะสมที่สุด โดยเป็นวิธีการที่เลียนแบบวิวัฒนาการของสิ่งมีชีวิต (Mitchell, 1998) ในการทำงานจะกระทำกับกลุ่มของโครโมโซม โดยต้องแปลงคำตอบของปัญหาให้อยู่ในรูปแบบโครโมโซมก่อน ขนาดของโครโมโซมจะแปรผันตามขนาดของปัญหา เมื่อโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหาก็จะมีขนาดใหญ่ขึ้นตามไปด้วย

ขั้นตอนวิธีเชิงพันธุกรรมเป็นการค้นหาแบบครอบคลุม (global search) คือ การค้นหาคำตอบจากปริภูมิการค้นหาสามารถกระโดดจากประชากรตัวหนึ่งไปอีกตัวหนึ่งที่ไม่จำเป็นต้องอยู่ใกล้กันได้ จึงทำให้สามารถหาคำตอบโดยรวมพบได้ง่าย แต่อาจจะส่งผลให้การค้นหาคำตอบที่ดีที่สุดเป็นไปได้ช้าเนื่องจากบางครั้งพบประชากรที่เป็นคำตอบที่ดีแล้วแต่ยังไม่ใช่คำตอบที่ดีที่สุดจึงกระโดดไปยังประชากรตัวอื่นในปริภูมิการค้นหาซึ่งอาจจะมีค่าความเหมาะสมน้อยกว่าตัวก่อนหน้าและอาจจะใช้เวลานาน กว่าจะกลับมายังตัวที่มีค่าความเหมาะสมที่ดีอีกครั้ง แสดงให้เห็นว่าในบางคำตอบที่เป็นคำตอบที่ดีจะมีส่วนที่ไม่ดีร่วมอยู่ด้วย จึงทำให้คำตอบนั้นยังไม่เป็นคำตอบที่ดีที่สุด

Genetic Local Search หรือ Memetic Algorithms

Memetic Algorithms (MAs) เป็น evolutionary algorithms (EAs) ที่รวมเทคนิคการค้นหาเฉพาะที่ (local search algorithm) เพื่อปรับปรุงประชากรแต่ละตัวให้ดีขึ้น (Fengjie, 2001) โดยวิธีการ Memetic Algorithms เป็นวิธีการแบบ population-based ที่ใช้ในการแก้ปัญหาการหาคำตอบที่เหมาะสมที่สุดซึ่งบางครั้งก็เรียกว่า Hybrid Genetic Algorithms (Moscato, 2002)

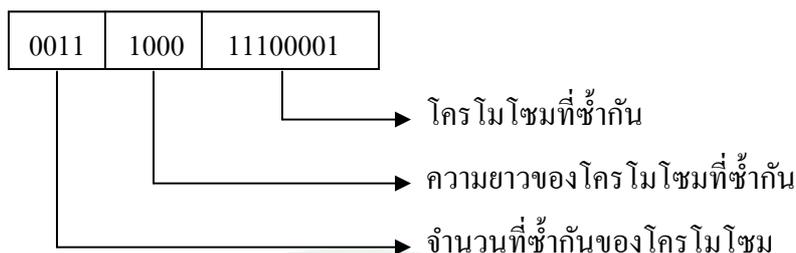
โดยงานวิจัยของ (Freisleben and Merz, 1996) ได้ปรับปรุงขั้นตอนวิธีเชิงพันธุกรรม โดยนำเทคนิคการค้นหาเฉพาะที่เข้ามาทำงานร่วมด้วยโดยทดสอบกับ Traveling Salesman Problem (TSP) ซึ่งคำตอบที่แปลงให้อยู่ในรูปแบบของโครโมโซมจะอยู่ในรูปการเรียงแบบสับเปลี่ยนของตัวเลข (combinatorial optimization) โดยจะเพิ่มส่วนของการค้นหาเฉพาะที่ในทุกครั้งที่ทำกระบวนการทางพันธุกรรม (การไขว้เปลี่ยนและการกลายพันธุ์) เพื่อเพิ่มประสิทธิภาพของการหาคำตอบที่ดีที่สุดเฉพาะที่ (local optima) ในปริภูมิการค้นหาและใช้ขั้นตอนวิธีเชิงพันธุกรรมเพื่อหา global optima จาก local optima นั้น โดยแสดงให้เห็นว่าประสิทธิภาพในการทำงานเพิ่มขึ้น

งานวิจัยของ (Nguyen and Duong, 2005) ได้นำขั้นตอนวิธีเชิงพันธุกรรมและเทคนิคการค้นหาเฉพาะที่แบบ hill-climbing มาใช้ในการจัดตารางสอบของมหาวิทยาลัย ซึ่งนำชุดข้อมูลจริงมาจากมหาวิทยาลัยโฮจิมิน โดยตัวปฏิบัติการที่ใช้มีเพียงการกลายพันธุ์ (mutation) เท่านั้น โดยผลที่ได้แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมผสมกับเทคนิคการค้นหาเฉพาะที่สามารถแก้ปัญหาการจัดตารางแบบ multi-objective ที่มีความยุ่งยากได้

เทคนิคการค้นหาเฉพาะที่ยังถูกนำไปใช้ในการทดสอบทางด้านคณิตศาสตร์ คือ (สุภักกานดาและคณะ, 2550) โดยใช้เทคนิคการค้นหาเฉพาะที่ที่เรียกว่า ชัฟเฟิลฟรอกลิปปีง (shuffled frog leaping) ได้นำวิธีดังกล่าวไปทดสอบในการแก้ปัญหาสมการคณิตศาสตร์ 2 สมการที่มีผลเฉลยแบบต่อเนื่อง และนำผลเฉลยที่ได้จากการทดสอบไปวิเคราะห์ค่าทางสถิติ และได้ข้อสรุปว่าการใช้เทคนิคชัฟเฟิลฟรอกลิปปีงใน Memetic Algorithm สามารถเพิ่มประสิทธิภาพของการประมาณค่าคำตอบที่ดีที่สุดได้ แต่ต้องใช้เวลาในการค้นหานั้นขึ้นกว่าเดิมเล็กน้อย

ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (Compressed Genetic Algorithm)

นักวิจัยอีกหลายท่านยังได้สังเกตเห็นข้อบกพร่องของขั้นตอนวิธีเชิงพันธุกรรมและทำการพัฒนาอย่างกว้างขวาง ได้มีการพัฒนาขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (Compressed Genetic Algorithm) ขึ้น โดยทดสอบกับปัญหา OneMax และปัญหาแขนหุ่นยนต์ โดยลดขนาดโครโมโซมลง เนื่องจากโครโมโซมปกติ จะเกิดจากโครโมโซมย่อยของข้อมูลที่ซ้ำกัน ดังนั้นจึงสามารถลดช่วงที่ซ้ำกันลงไปได้ โดยภายในโครโมโซมให้เก็บเพียงแค่ความยาวของโครโมโซมที่ซ้ำกัน จำนวนที่ซ้ำกันของโครโมโซม และ โครโมโซมที่ซ้ำกัน (วรเศรษฐ, 2551) ดังภาพที่ 11



ภาพที่ 11 ตัวอย่างโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (Compressed Genetic Algorithm)

โดยขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดนี้จะมีการทำงานที่แตกต่างจากขั้นตอนวิธีเชิงพันธุกรรม คือก่อนจะนำโครโมโซมไปวัดค่าความเหมาะสม ต้องมีการคลายโครโมโซมก่อน จากภาพที่ 2 จะได้ว่าโครโมโซม 11100001 ซ้ำกัน 3 ครั้ง โดยโครโมโซมมีความยาวเท่ากับ 8 ดังนั้นเมื่อนำโครโมโซมชุดนี้มาคลายจะมีค่าเป็น 111000011110000111100001

ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดนี้ทำให้ปริภูมิการค้นหามีขนาดเล็กลง เนื่องจากโครโมโซมมีขนาดเล็กลง จึงสามารถทำงานได้เร็วกว่าขั้นตอนวิธีเชิงพันธุกรรมและยังใช้หน่วยความจำน้อยกว่า แต่ก็ยังไม่สามารถแก้ปัญหาได้กับทุกปัญหาซึ่งในบางปัญหาที่อาจจะหาคำตอบไม่พบและยังมีข้อจำกัดด้านอัตราการบีบอัดข้อมูล

ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม LZW (LZWGA)

ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดถูกพัฒนาขึ้นโดยนำอัลกอริทึม Lempel-Ziv-Welch (LZW) เข้ามาใช้ และได้ทดสอบกับปัญหา OneMax โดยข้อมูลที่เก็บในโครโมโซมจะเปลี่ยนเป็นตัวเลขจำนวนเต็ม ซึ่งในการสร้างประชากรเริ่มต้นจะต่างจากขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดคือประชากรรุ่นแรกของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด ที่ถูกสร้างแบบสุ่มจะสามารถนำมาคลายได้ทุกกรณีไม่ว่าจะสุ่มรูปแบบใดก็ตาม ส่วนประชากรของ LZWGA จะต้องสุ่มมาภายใต้เงื่อนไขบางอย่างเพื่อให้สามารถคลายได้ด้วยอัลกอริทึมการคลายของ LZW (นริศ , 2548) จากผลการทดลองแสดงให้เห็นว่า LZWGA สามารถหาคำตอบได้เร็วกว่า CGA ที่ปัญหาที่มีขนาดใหญ่ตั้งแต่ 1024 บิตขึ้นไป

การทดสอบประสิทธิภาพของ LZWGA

เพื่อเป็นการทดสอบประสิทธิภาพของ LZWGA ผู้วิจัยจึงได้นำเสนอปัญหาเพื่อนำมาทดสอบประสิทธิภาพของ LZWGA (สาวตรี, 2550) โดยมีสมมติฐานที่ว่า หากอัลกอริทึมนี้มีประสิทธิภาพดีจริง เมื่อนำไปทดสอบกับปัญหาใด ประสิทธิภาพย่อมไม่ลดลงไป โดยปัญหาที่นำเสนอจะมีพื้นฐานมาจากปัญหาพื้นฐาน แต่ใช้การสุ่มคำตอบเพื่อให้ได้คำตอบที่ไม่มีแบบแผน ปัญหาที่นำเสนอคือ ปัญหา RandomMax และ ปัญหา RandomRoyalRoad แต่จากผลการทดลองที่ได้ไม่เป็นไปตามสมมติฐาน กล่าวคือ อัลกอริทึม LZWGA ไม่สามารถแก้ปัญหา RandomMax และ ปัญหา RandomRoyalRoad ได้ และยังมีประสิทธิภาพในการทำงานต่ำกว่า GA ทั่วไป โดยอาจเนื่องมาจากรูปแบบของค่าในโครโมโซมไม่มีแบบแผน เพราะค่าทุกบิตในโครโมโซมเกิดจากการสุ่ม โดยประสิทธิภาพของ LZWGA จะดีในช่วงแรก ๆ และลดลงในภายหลัง

ในงานวิจัยชิ้นนี้ ได้พยายามที่จะปรับปรุงประสิทธิภาพของ LZWGA เพื่อให้สามารถแก้ปัญหาที่กล่าวข้างต้นได้ จึงได้นำวิธีการค้นหาแบบ local search เข้ามาร่วมกับ LZW เรียกว่า Hybrid LZWGA

อุปกรณ์และวิธีการ

อุปกรณ์

1. Hardware:

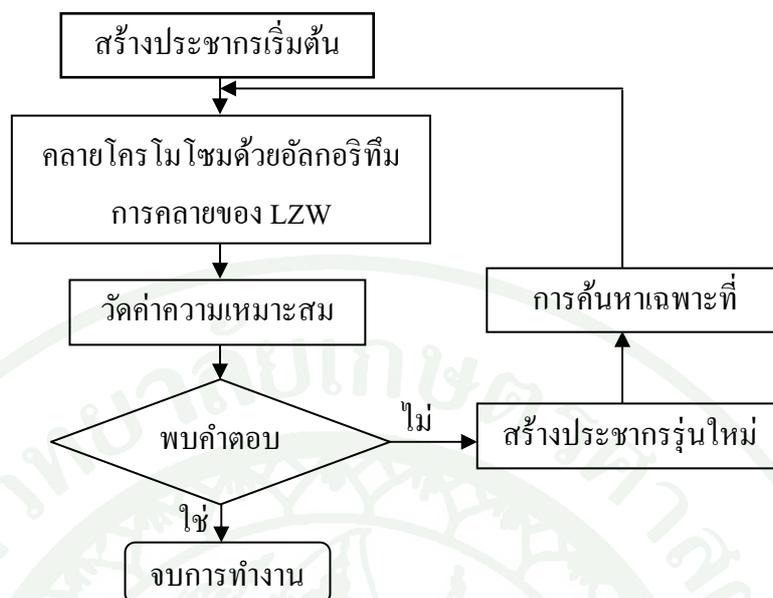
- 1.1 เครื่องคอมพิวเตอร์ส่วนบุคคล Intel® Core™2 Quad 2.4 GHz
หน่วยความจำ 2 GB

2. Software:

- 2.1 Java Development Toolkit: JDK1.6
- 2.2 NetBeans IDE 6.1
- 2.3 Operating System: Microsoft Windows XP

วิธีการ

งานวิจัยนี้ได้นำอัลกอริทึมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดแบบ LZW มาใช้ร่วมกับวิธีการค้นหาเฉพาะที่ เรียกว่า ขั้นตอนวิธีเชิงพันธุกรรมแบบผสม (Hybrid LZWGA) โดยจะนำการค้นหาเฉพาะที่มาใช้หลังการทำกระบวนการทางพันธุกรรม (genetic operator) แล้ว เพื่อหวังจะเพิ่มประสิทธิภาพของการทำงานให้ดีขึ้น โดยมีขั้นตอนการทำงาน ดังภาพที่ 12



ภาพที่ 12 ขั้นตอนการทำงานของ Hybrid LZWGA

1. ลักษณะของปัญหาที่ใช้ในการทดลอง

ปัญหาที่ใช้ในการทดลองเป็นปัญหาแบบสุ่ม ได้แก่

ปัญหา RandomMax

ปัญหา OneMax คือ ปัญหาการนับจำนวนเลขหนึ่งในโครโมโซม โดยค่าความเหมาะสมที่ได้ก็คือ จำนวนเลขหนึ่งทั้งหมดที่พบในโครโมโซม ดังนั้นโครโมโซมที่ดีที่สุดคือโครโมโซมที่ทุกบิตมีค่าเป็นหนึ่ง ปัญหา OneMax จำลองปัญหาการหาค่าเหมาะสมที่สุดที่ตัวแปรทุกตัวเป็นอิสระต่อกัน

ปัญหา RandomMax คือ ปัญหาการนับจำนวนบิตที่ตรงกับคำตอบในโครโมโซม โดยค่าความเหมาะสมที่ได้คือ จำนวนบิต ณ ตำแหน่งเดียวกับคำตอบมีค่าภายในตรงกัน ซึ่งคำตอบจะเกิดจากการสุ่ม เช่น

คำตอบ : 10100101

โครโมโซม : 10100100

ค่าความเหมาะสมที่ได้จะเป็นค่าของบิตในตำแหน่งเดียวกันที่มีค่าเท่ากันคือ 7 โดยอัลกอริทึมในการทำงานจะมี 2 ส่วนคือ ส่วนของการสุ่มคำตอบและส่วนของการวัดค่าความเหมาะสม ดังแสดงในภาพที่ 13

ค่าความสุ่ม

ในการสร้างคำตอบที่เกิดจากสุ่มนั้น จะมีการกำหนดค่าความสุ่ม ซึ่งในการทดลองจะมีสองค่าคือ 0 และ 1 โดยค่าความสุ่มเท่ากับ 0 คือไม่มีการสุ่มเกิดขึ้นเลย (คำตอบที่ได้เป็น 1 หหมด) และค่าความสุ่มเท่ากับ 1 คือทุกบิตของโครโมโซมจะเกิดจากการสุ่มหมด

```

CreateTargetChromosome()
begin
  for i=0 to chromosomeLength-1
    target[i]←random(0 or1)
  end for
end

int FitnessEvaluate(int[] chromosome)
begin
  fitness←0
  for i=0 to chromosomeLength-1
    if chromosome[i]=target[i]
      fitness++
    end if
  end for
  return fitness
end

```

ภาพที่ 13 รหัสเทียมของ RandomMax กรณีที่ค่าความสุ่มเป็น 1

ปัญหา RandomRoyalRoad

ปัญหา RoyalRoad เป็นปัญหาที่จำลองปัญหาที่มีความสัมพันธ์ของบิตในโครโมโซม โดยคำตอบที่ต้องการคือค่าของบิตทุกตัวในบล็อกมีค่าเท่ากับ 1 ในการวัดค่าความเหมาะสมหากบิตใด

บิตหนึ่งในบล็อกมีค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็นศูนย์ ในทางกลับกัน หากทุกบิตในบล็อก มีค่าเป็น 1 หมด จะได้ค่าความเหมาะสมเท่ากับจำนวนบิตในบล็อกนั้น

ปัญหา RandomRoyalRoad แตกต่างจาก RoyalRoad โดย ถ้าค่าของบิตในบล็อกนั้นทุกค่า มีค่าเหมือนกับคำตอบซึ่งเกิดจากการสุ่ม ความเหมาะสมที่ได้เท่ากับจำนวนบิตในบล็อก แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่เหมือนคำตอบ บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันที เช่น

คำตอบ : 1110 1110 1110 1110

โครโมโซม : 1011 1110 1001 1110

ค่าความเหมาะสมที่ได้จะเท่ากับ 8 เนื่องจากในบล็อกที่ 2 และ 4 โครโมโซมมีค่าเท่ากับคำตอบทุกตำแหน่งส่วนบล็อกที่ 1 และ 3 จะไม่ได้คะแนนเนื่องจากมีบางบิตในโครโมโซมที่มีค่าไม่ตรงกับคำตอบ โดยอัลกอริทึมของ RandomRoyalRoad จะมีการทำงานหลัก 2 ส่วนคือ ส่วนของการสุ่มบล็อกและส่วนของการวัดค่าความเหมาะสม ดังแสดงในภาพที่ 14

การทำงาน

ในส่วนของการสร้างคำตอบแบบสุ่ม (CreateTargetChromosome()) จะสุ่มตัวเลข 0 หรือ 1 จำนวนเท่ากับขนาดของบล็อกที่กำหนดไว้ เช่น บล็อกขนาดเท่ากับ 4 จะสุ่มตัวเลขมา 4 ตัว แล้วนำตัวเลขทั้งสี่ไปคัดลอกใส่บล็อกอื่น ๆ จนความยาวทั้งหมดเท่ากับความยาวของโครโมโซมที่กำหนดไว้

ส่วนของการวัดค่าความเหมาะสม (FitnessEvaluate()) มีการรับโครโมโซมที่จะนำมาเปรียบเทียบกับคำตอบที่สุ่มไว้ แล้วเปรียบเทียบบิตต่อบิต ถ้าค่าของบิตของโครโมโซมและคำตอบตรงกัน จะได้ค่าความเหมาะสมเท่ากับ 1

```

CreateTargetChromosome()
begin
  for i=0 to blockSize-1
    block[i] ← random(0 or 1)
  end for
  m ← 0
  for j=0 to chromosomeLength-1
    for k=0 to blockSize-1
      target[m] ← block[k]
      m++
    end for
    j ← j+blockSize
  end for
end

int FitnessEvaluate(int[] chromosome)
begin
  count ← 0, fitness ← 0, k ← 0
  for i=0 to chromosomeLength-1
    for j=0 to blockSize-1
      if chromosome[k]=target[k]
        count++
        k++
      end if
    end for
    if count=blockSize
      fitness++
    end if
    i ← i+blockSize
  end for
  return fitness*blockSize
end

```

ภาพที่ 14 รหัสเทียมของ RandomRoyalRoad กรณีที่ค่าความสุ่มเป็น 1

2. แนวทางการทดลอง

2.1 ทดสอบประสิทธิภาพของ LZWGA กับ GA

เราเปรียบเทียบประสิทธิภาพของ LZWGA และ GA โดยทดลองกับปัญหา RandomMax และ RandomRoyalRoad ขนาดหนึ่งพันบิต ที่โครโมโซม LZWGA ขนาด 168 บิต ค่าพารามิเตอร์ที่ใช้ในแต่ละอัลกอริทึมเป็นค่าที่ทำให้อัลกอริทึมได้ผลดีจากการทดลองเบื้องต้น ทั้งนี้เพราะว่าค่าพารามิเตอร์มีผลต่อคำตอบที่ได้ โดยค่าพารามิเตอร์ที่ใช้ได้แสดงในตารางที่ 1 และ 2

เนื่องจากอัลกอริทึมที่ใช้ในการทดลองนี้เป็นอัลกอริทึมที่มีการสุ่มตัวเลข จึงทดลอง 10 ครั้งแล้วหาค่าเฉลี่ย โดยปัญหาที่ทดลองคือ

1. เปรียบเทียบประสิทธิภาพของ LZWGA กับ GA ในการแก้ปัญหา RandomMax ที่ค่าความสุ่มเป็น 0 และ 1
2. ประสิทธิภาพของ LZWGA กับ GA ในปัญหา RandomRoyalRoad ที่ค่าความสุ่มเป็น 0 และ 1 โดยกำหนดให้ขนาดบิตคือเป็น 4 และ 8

ตารางที่ 1 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomMax

พารามิเตอร์	LZWGA	GA
ขนาดประชากร	1000 ตัว	
ค่าความเหมาะสมของคำตอบ	1,000	
วิธีการคัดเลือก	แบบประลอง โครโมโซมที่ประลองกันมี 4 ตัว	
วิธีการไขว้เปลี่ยน	No crossover	Uniform
อัตราการไขว้เปลี่ยน	0%	80%
อัตราการกลายพันธุ์	1%	0%
เก็บโครโมโซมที่ดีที่สุดไว้	10 ตัว	
จำนวนรุ่นจำกัดไว้ที่	200 รุ่น	

ตารางที่ 2 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad

พารามิเตอร์	LZWGA	GA
ขนาดประชากร	1000 ตัว	
ค่าความเหมาะสมของคำตอบ	1,000	
วิธีการคัดเลือก	แบบประลอง โครโมโซมที่ประลองกันมี 4 ตัว	
อัตราการไขว้เปลี่ยน	80%	80%
วิธีการไขว้เปลี่ยน	Single point	Uniform
อัตราการกลายพันธุ์	1%	0%
เก็บโครโมโซมที่ดีที่สุดไว้	10 ตัว	
จำนวนรุ่นจำกัดไว้ที่	200 รุ่น	

2.2 ทดสอบประสิทธิภาพของ LZWGA กับ Hybrid LZWGA

2.2.1 เปรียบเทียบประสิทธิภาพของ LZWGA และ Hybrid LZWGA โดยทดลองกับปัญหา RandomMax และ RandomRoyalRoad ที่โครโมโซม LZWGA ขนาดสองร้อยบิต โดยโครโมโซม Hybrid LZWGA จะทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดขณะบีบอัดอยู่เท่านั้น

ค่าพารามิเตอร์ที่ใช้เป็นค่าที่อ้างอิงจากการทดลองเบื้องต้นของ LZWGA โดยจำกัดจำนวนรุ่นไว้ที่ 200 รุ่น ค่าความเหมาะสมที่เป็นไปได้คือ ค่าความเหมาะสมที่เท่ากับ 1000 โดยใช้ค่าพารามิเตอร์เดียวกับตารางที่ 1 และ 2 เฉพาะในส่วนของ LZWGA

2.3 ทดสอบประสิทธิภาพระหว่าง Hybrid LZWGA โดยแบ่งออกเป็น 4 กรณี คือ

2.3.1 Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดขณะบีบอัดอยู่

2.3.2 Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดขณะคลายแล้ว

2.3.3 Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับทุกโครโมโซม ขณะบีบอัดอยู่

2.3.4 Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับทุกโครโมโซม ขณะคลายแล้ว

เปรียบเทียบประสิทธิภาพเพื่อหาว่าอัลกอริทึม Hybrid LZWGA แบบใด ที่ให้คำตอบที่ดีที่สุด โดยทดลองกับปัญหา RandomMax และ RandomRoyalRoad ที่โครโมโซม LZWGA ขนาดสองร้อยบิต ค่าพารามิเตอร์ที่ใช้เป็นค่าที่อ้างอิงจากการทดลองเบื้องต้นของ LZWGA จำกัดจำนวนรุ่นไว้ที่ 200 รุ่น จำนวนประชากรที่ใช้ 1,000 ตัว ค่าความเหมาะสมที่เป็นไปได้คือ ค่าความเหมาะสมที่เท่ากับ 1000 โดยค่าพารามิเตอร์ที่ใช้ เท่ากับค่าพารามิเตอร์ในตารางที่ 1 และ 2 ในส่วนของ LZWGA

2.4 เปรียบเทียบประสิทธิภาพของ Hybrid LZWGA แบบที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัวขณะคลายแล้ว และ Hybrid GA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัว โดยทดลองกับปัญหา RandomMax และ RandomRoyalRoad ที่โครโมโซม LZWGA ขนาดสองร้อยบิต โครโมโซม Hybrid GA ขนาดหนึ่งพันบิต

ค่าพารามิเตอร์ที่ใช้เป็นค่าที่อ้างอิงจากการทดลองเบื้องต้นของ LZWGA และ GA จำกัดจำนวนรุ่นไว้ที่ 200 รุ่น ค่าความเหมาะสมสูงสุดคือ ค่าความเหมาะสมที่เท่ากับ 1000 โดย Hybrid LZWGA ใช้ค่าพารามิเตอร์เดียวกับตารางที่ 1 และ 2 ในส่วนของ LZWGA และ Hybrid GA ใช้ค่าพารามิเตอร์เดียวกับตารางที่ 1 และ 2 ในส่วนของ GA

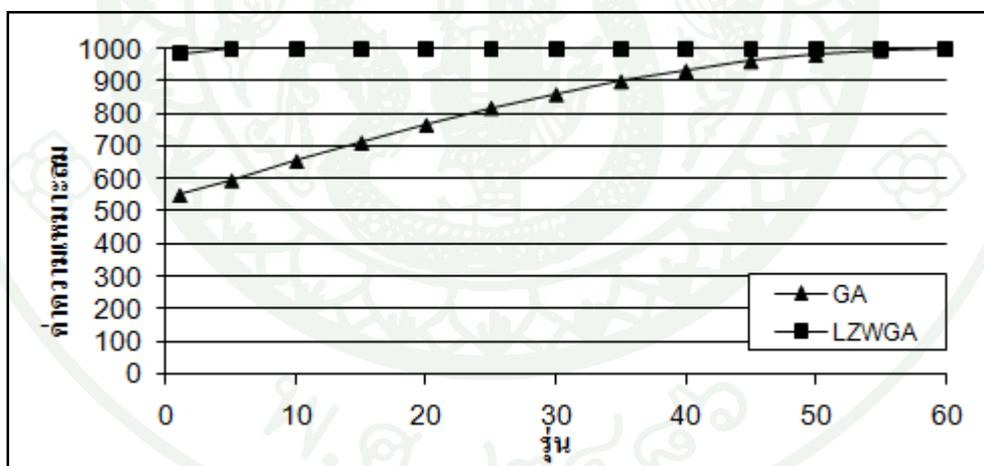
ผลและวิจารณ์

ผล

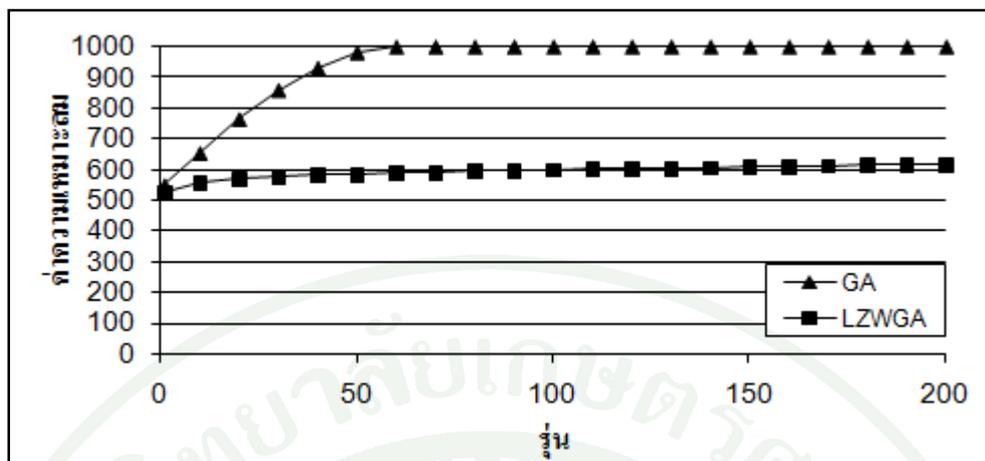
1. ผลการทดลองเปรียบเทียบประสิทธิภาพของ LZWGA กับ GA

1.1 ปัญหา RandomMax

การเปรียบเทียบประสิทธิภาพของ LZWGA และ GA ในปัญหา RandomMax ที่ค่าความสุ่มเป็น 0 (ภาพที่ 15) แสดงให้เห็นว่า LZWGA ค้นพบคำตอบได้เร็วกว่า GA มาก ซึ่งสอดคล้องกับงานวิจัย (Kunasol, 2006) แต่เมื่อเปลี่ยนค่าความสุ่มเป็น 1 พบว่า LZWGA จะหาคำตอบไม่พบ (ภาพที่ 16) เนื่องจาก LZWGA จะมีประสิทธิภาพดีเมื่อทำงานกับโครโมโซมที่มีแบบแผน ในขณะที่ประสิทธิภาพของ GA ไม่เปลี่ยนแปลงถึงแม้ค่าความสุ่มจะเปลี่ยนจาก 0 เป็น 1



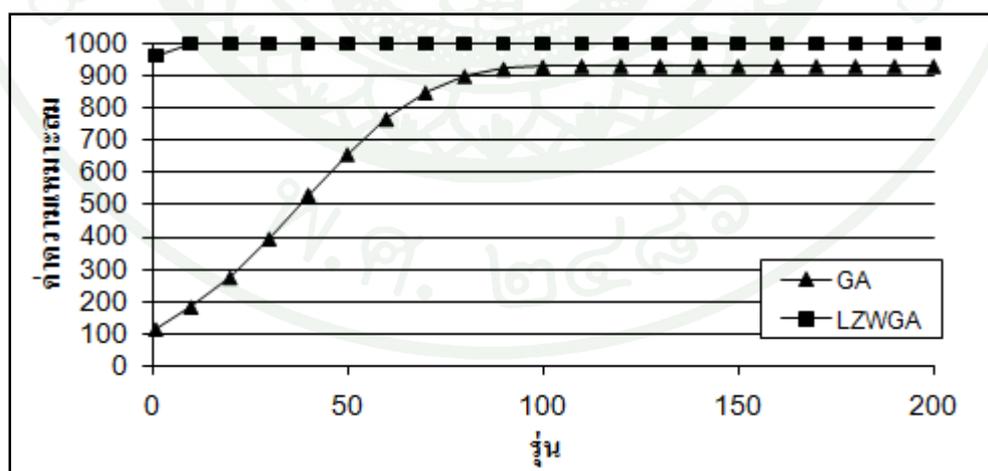
ภาพที่ 15 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 0



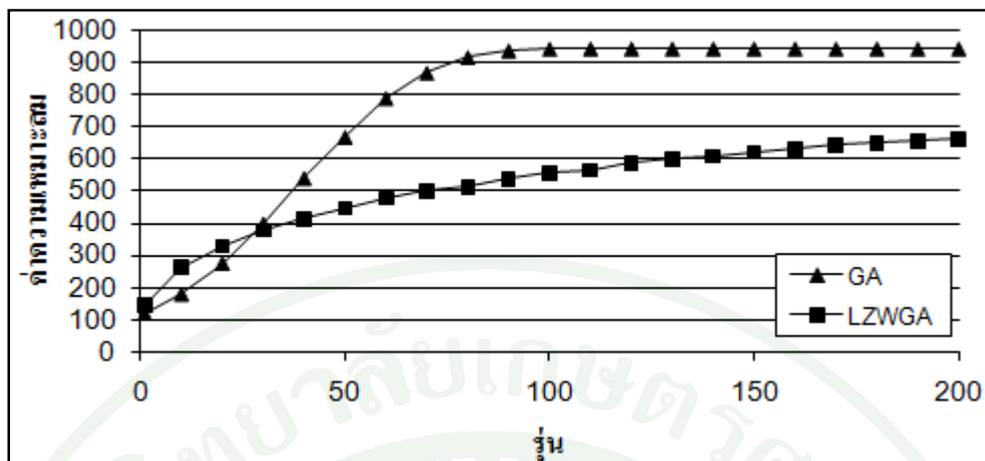
ภาพที่ 16 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 1

1.2 ปัญหา RandomRoyalRoad

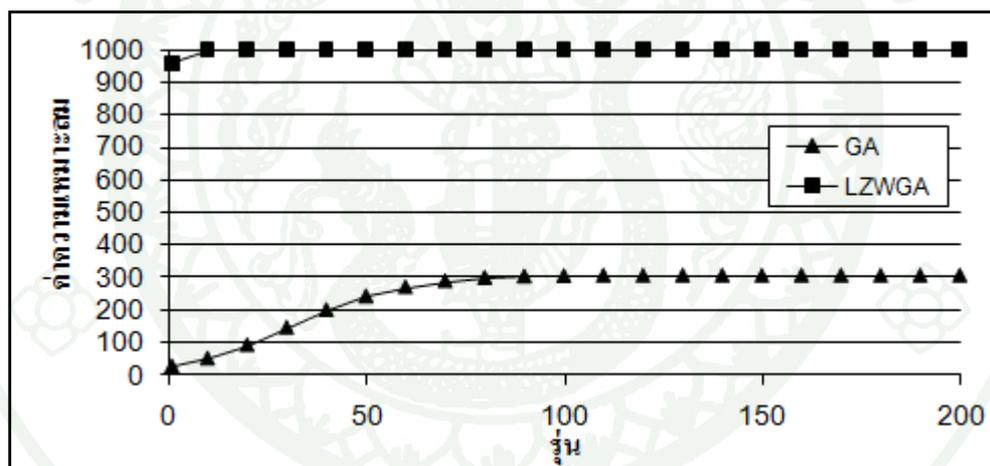
ปัญหา RandomRoyalRoad ที่ค่าความสุ่มเป็น 0 และขนาดบล็อกเป็น 4 และ 8 (ภาพที่ 17 และ 19) LZWGA สามารถหาคำตอบได้เร็วกว่า GA แต่เมื่อเปลี่ยนค่าความสุ่มเป็น 1 (ภาพที่ 20 และ 22) ทั้ง LZWGA และ GA ไม่สามารถหาคำตอบพบ



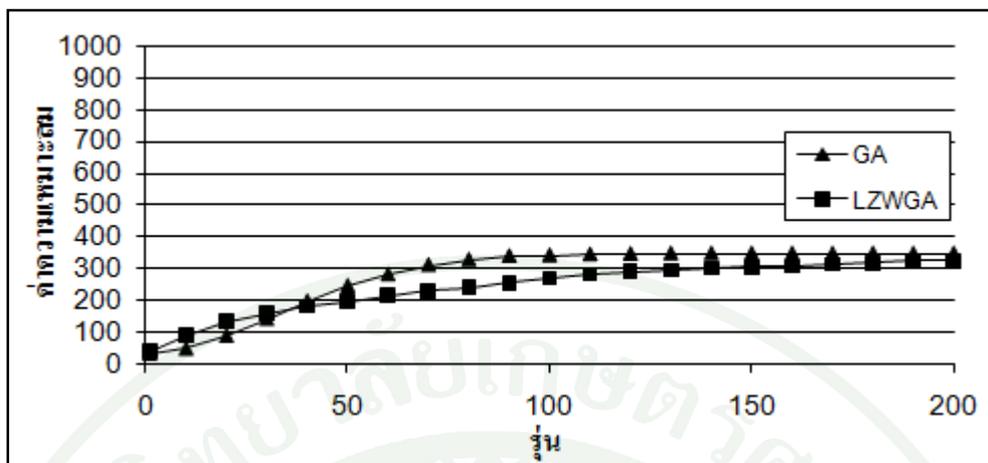
ภาพที่ 17 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 0 และขนาดบล็อกเท่ากับ 4



ภาพที่ 18 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 1 และขนาดบล็อกลูกเท่ากับ 4



ภาพที่ 19 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 0 และขนาดบล็อกลูกเท่ากับ 8

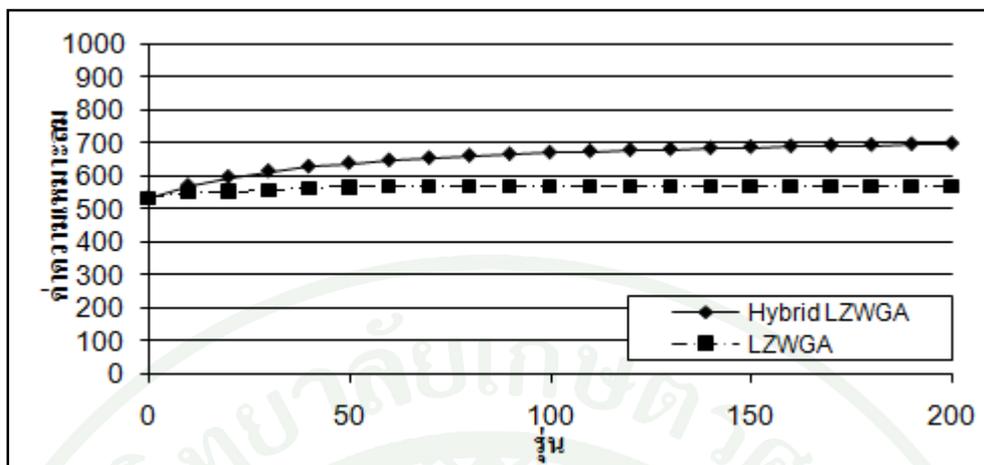


ภาพที่ 20 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง GA และ LZWGA ที่ค่าความสุ่มเท่ากับ 1 และขนาดบล็อกลูกเท่ากับ 8

2. ผลการทดลองเปรียบเทียบประสิทธิภาพของ LZWGA กับ Hybrid LZWGA

2.1 ปัญหา RandomMax

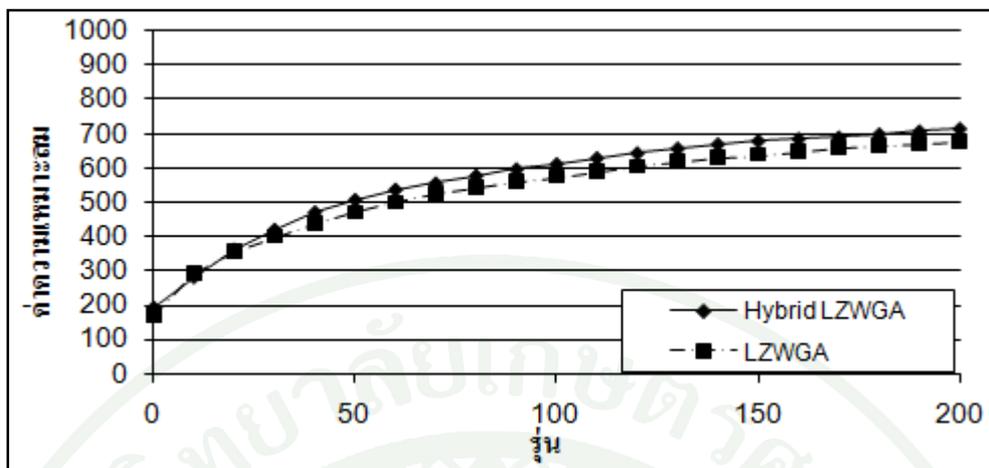
การเปรียบเทียบประสิทธิภาพของ LZWGA และ Hybrid LZWGA ค่าความเหมาะสมสูงสุดที่เป็นไปได้คือ 1000 โดยโครโมโซม Hybrid LZWGA จะทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดขณะบิบอัดอยู่เท่านั้น จากผลการทดลอง แสดงให้เห็นว่า Hybrid LZWGA มีประสิทธิภาพในการทำงานในปัญหา RandomMax (ภาพที่ 21) ดีขึ้นกว่า LZWGA โดยประสิทธิภาพในการทำงานดีขึ้นประมาณ 22 เปอร์เซ็นต์



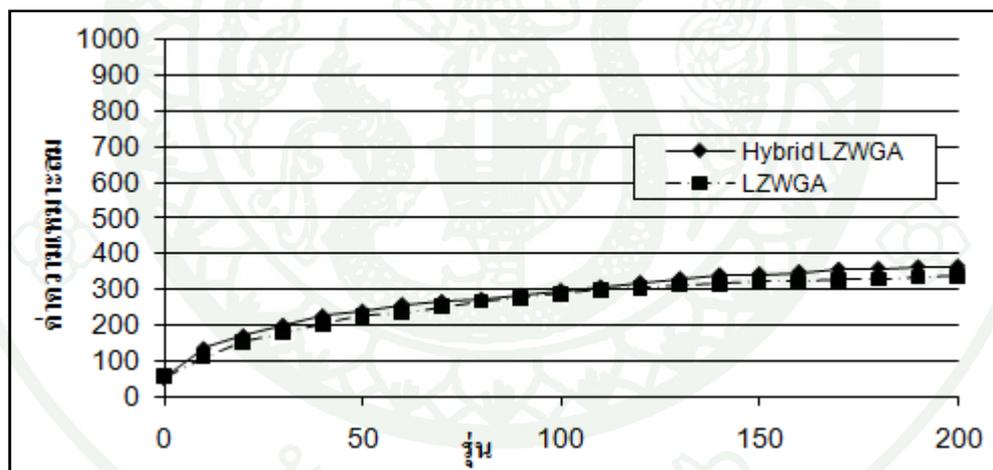
ภาพที่ 21 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดที่บีบอัดอยู่

2.2 ปัญหา RandomRoyalRoad

ปัญหา RandomRoyalRoad ที่ขนาดบล็อกเป็น 4 และ 8 (ภาพที่ 22 และ 23) Hybrid LZWGA สามารถหาคำตอบได้เร็วกว่า LZWGA เล็กน้อย โดยประสิทธิภาพดีขึ้นประมาณ 5 และ 7 เปอร์เซ็นต์ ตามลำดับ แต่อย่างไรก็ตาม ทั้ง Hybrid LZWGA และ LZWGA ก็ยังไม่สามารถหาคำตอบของปัญหาแบบสุ่มได้



ภาพที่ 22 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดที่มีบิตอยู่ ที่ขนาดบล็อกเท่ากับ 4



ภาพที่ 23 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดที่มีบิตอยู่ ที่ขนาดบล็อกเท่ากับ 8

เนื่องจากการเปรียบเทียบผลการทำงานของ LZWGA และ Hybrid LZWGA โดยใช้กราฟเป็นตัวตัดสินนั้น อาจยังไม่ชัดเจนถึงความแตกต่างที่พบ จึงได้นำวิธีการทางสถิติคือ การทดสอบแบบ T-Test มาใช้ในการวัดความแตกต่างของประสิทธิภาพการทำงานระหว่าง LZWGA และ Hybrid LZWGA โดยใช้การทดสอบแบบ Paired Samples (กรณีในกลุ่มตัวอย่างทั้งสองไม่อิสระต่อ

กัน) เนื่องจากในการทดลองได้ใช้ค่าพารามิเตอร์ที่เหมือนกันทุกประการ (ตารางที่ 1 และ 2) ดังตารางที่ 3 โดยเปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ในปัญหา RandomMax (RM) และ RandomRoyalRoad (RRR) ที่ขนาดบล็อกเป็น 4 และ 8 ตามลำดับ จากตารางที่ 3 $t = 11.739$, $t = 11.953$ และ $t = 9.877$ มีค่านัยสำคัญ (Sig.) เป็น 0.000 ซึ่งน้อยกว่า 0.05 ปฏิเสธ H_0 นั่นคือ Hybrid LZWGA และ LZWGA มีความแตกต่างกันอย่างมีนัยสำคัญที่ระดับ 0.05

ตารางที่ 3 การทดสอบความแตกต่างของค่าเฉลี่ยกรณีในกลุ่มตัวอย่างทั้งสองไม่อิสระต่อกัน โดยใช้วิธี T-Test

HYBRID LZWGA เปรียบเทียบ กับ LZWGA	กรณีในกลุ่มตัวอย่างทั้งสองไม่อิสระต่อกัน					t	df	Sig. (2- tailed)
	ค่าเฉลี่ย	ค่าเบี่ยงเบน มาตรฐาน	ค่าคลาดเคลื่อน เฉลี่ย มาตรฐาน	ช่วงความเชื่อมั่น 95%				
				ค่า ต่ำสุด	ค่า สูงสุด			
RM	90.933	35.4969	7.746	74.775	107.091	11.739	20	0.000
RRR block = 4	31.638	12.1294	2.646	26.116	37.159	11.953	20	0.000
RRR block = 8	18.743	8.696	1.898	14.785	22.701	9.877	20	0.000

df = องศาอิสระ (degree of freedom)

2-tail = การทดสอบแบบสองทาง

จากการทดลอง แสดงให้เห็นว่า ประสิทธิภาพในการหาคำตอบของ Hybrid LZWGA เมื่อเปรียบเทียบกับ LZWGA มีประสิทธิภาพดีขึ้น นั่นหมายความว่า อัลกอริทึมในการค้นหาเฉพาะที่ที่เพิ่มเข้าไป มีผลทำให้ประสิทธิภาพโดยรวมในการทำงานของอัลกอริทึม LZWGA ดีขึ้น ดังนั้น จึงทำการปรับปรุงเฉพาะในส่วนของการค้นหาเฉพาะที่ จากเดิมที่ใช้การค้นหาเฉพาะที่แบบ hill

climbing โดยปรับปรุงเฉพาะค่าที่ดีที่สุดที่อยู่ในรูปโครโมโซมแบบบิปปัด จะเพิ่มกรณีในการทำการค้นหาเฉพาะที่อีก 3 แบบคือ

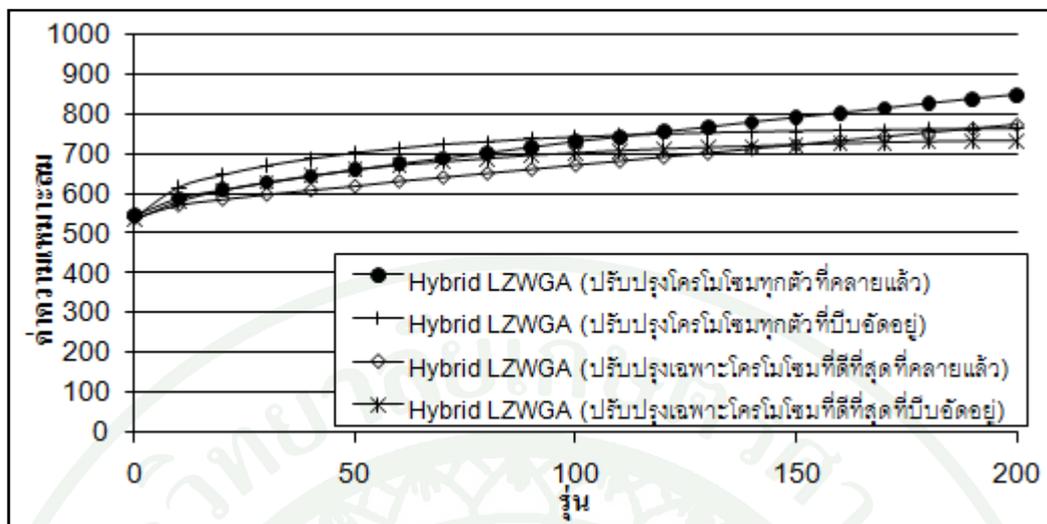
- 1) Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับทุกโครโมโซม ขณะที่บิปปัดอยู่
- 2) Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับทุกโครโมโซม ขณะที่คลายแล้ว
- 3) Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะตัวที่ดีที่สุด ขณะที่คลายแล้ว

ทำการทดลองกับปัญหา RandomMax และ RandomRoyalRoad ที่ขนาดบิตคือเป็น 4 และ 8 ได้ผลการทดลองดังนี้

3. ผลการทดลองเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA 4 แบบ

3.1 ปัญหา RandomMax

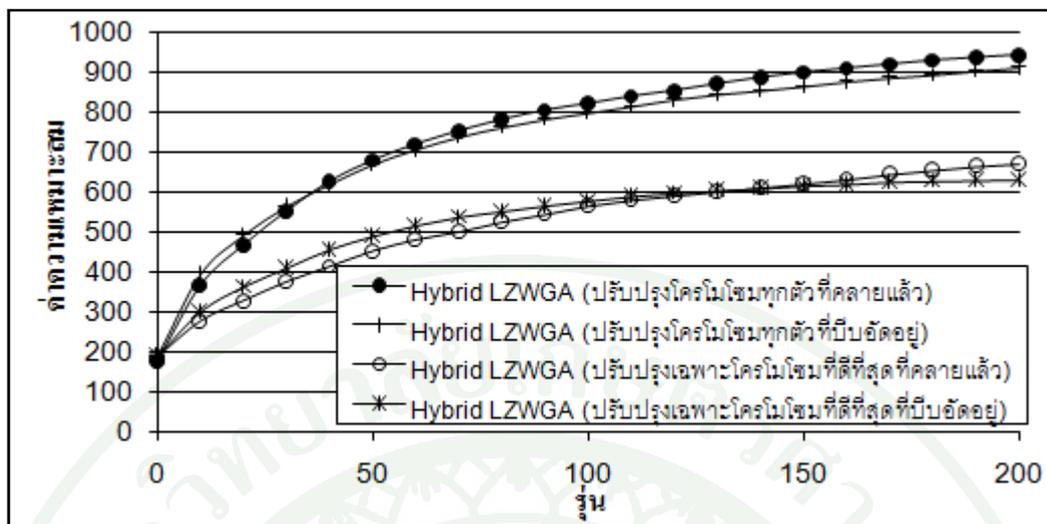
การเปรียบเทียบประสิทธิภาพของ Hybrid LZWGA ทั้ง 4 แบบ ค่าความเหมาะสมที่เป็นไปได้คือ ค่าความเหมาะสมที่เท่ากับ 1000 จากผลการทดลอง แสดงให้เห็นว่า Hybrid LZWGA แบบที่นำการค้นหาเฉพาะที่มาปรับปรุงโครโมโซมทุกตัวขณะที่คลายแล้ว มีประสิทธิภาพในการทำงานในปัญหา RandomMax (ภาพที่ 24) ดีขึ้นกว่าแบบเดิม (ปรับปรุงเฉพาะตัวที่ดีที่สุด ขณะที่บิปปัดอยู่) ประมาณ 16 เปอร์เซ็นต์ และดีที่สุด ใน 4 แบบที่นำมาทดลอง



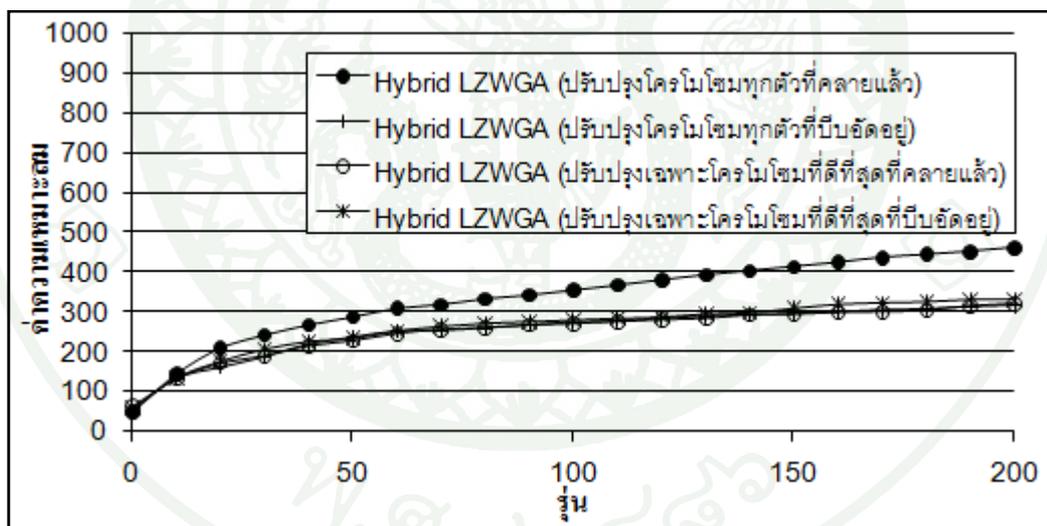
ภาพที่ 24 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง Hybrid LZWGA 4 แบบ

3.2 ปัญหา RandomRoyalRoad

ปัญหา RandomRoyalRoad ที่ขนาดบล็อกรุ่นเป็น 4 และ 8 (ภาพที่ 25 และ 26) Hybrid LZWGA แบบที่นำการค้นหาเฉพาะที่ปรับปรุงโครโมโซมทุกตัวขณะที่คลายแล้ว มีประสิทธิภาพในการทำงานในปัญหา RandomRoyalRoad ดีขึ้นกว่าแบบเดิม (ปรับปรุงเฉพาะตัวที่ดีที่สุด ขณะที่บีบอัดอยู่) ประมาณ 50 เปอร์เซ็นต์ และ 40 เปอร์เซ็นต์ ตามลำดับ และดีที่สุด ใน 4 แบบที่นำมาทดลอง



ภาพที่ 25 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA 4 แบบ ที่ขนาดบล็อกละเท่ากับ 4



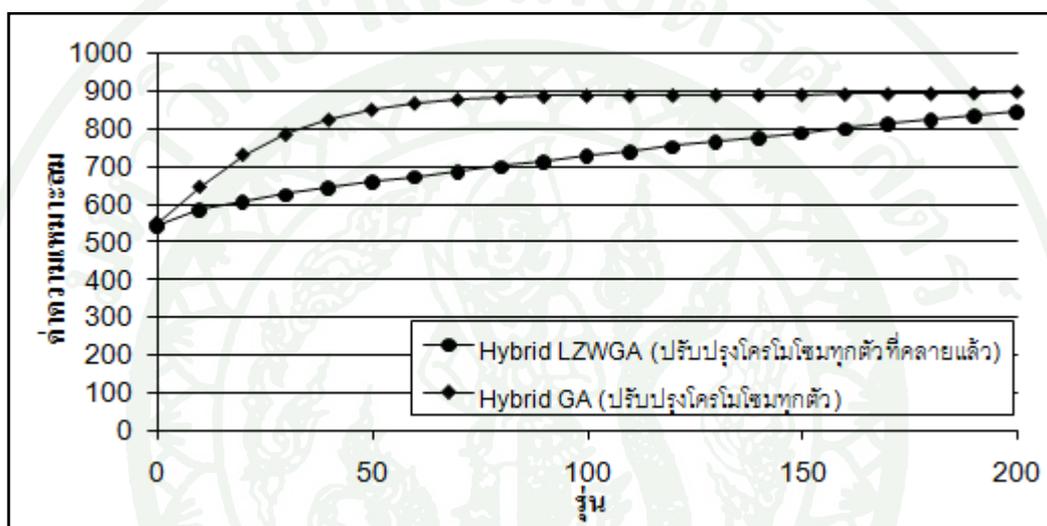
ภาพที่ 26 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA 4 แบบ ที่ขนาดบล็อกละเท่ากับ 8

จากกราฟที่ 24, 25 และ 26 แสดงให้เห็นว่า Hybrid LZWGA แบบที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัวที่คลายแล้ว ให้ค่าความเหมาะสมมากที่สุด ซึ่งการทำการค้นหาเฉพาะที่กับโครโมโซมที่คลายแล้วมีความหมายใกล้เคียงกับการนำโครโมโซมของ GA ทุกตัวมาทำการค้นหาเฉพาะที่ ดังนั้น ผู้วิจัยจึงทำการเปรียบเทียบประสิทธิภาพในการหาคำตอบระหว่าง Hybrid LZWGA

แบบที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัวที่คล้ายแล้วกับ GA ที่ทำการค้นหาเฉพาะที่ (Hybrid GA) กับโครโมโซมทุกตัว และแสดงเป็นกราฟตามภาพที่ 27, 28 และ 29

4. ผลการทดลองเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA 4 แบบ

4.1 ปัญหา RandomMax



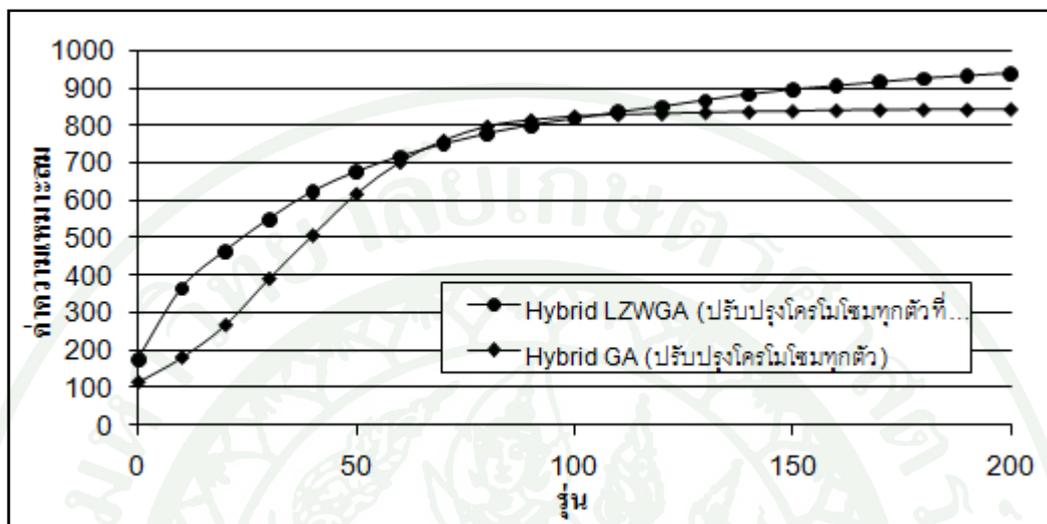
ภาพที่ 27 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax เปรียบเทียบระหว่าง Hybrid LZWGA และ Hybrid GA

ประสิทธิภาพของ Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัวขณะคล้ายแล้ว เมื่อเปรียบเทียบกับ Hybrid GA ค่าความเหมาะสมสูงสุดคือ ค่าความเหมาะสมที่เท่ากับ 1000 จากผลการทดลอง แสดงให้เห็นว่า Hybrid LZWGA มีประสิทธิภาพในการทำงานในปัญหา RandomMax (ภาพที่ 27) ดีกว่า Hybrid GA

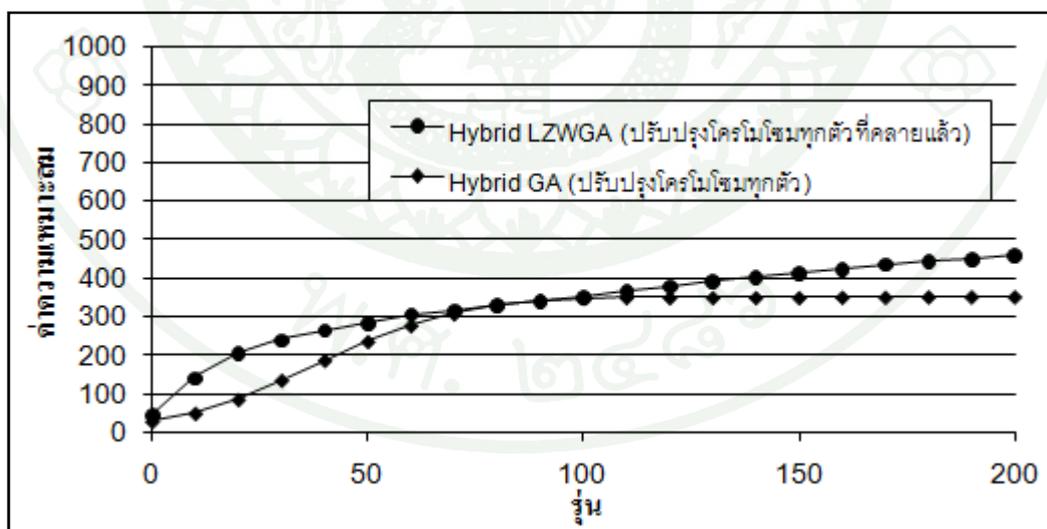
4.2 ปัญหา RandomRoyalRoad

ปัญหา RandomRoyalRoad ที่ขนาดบล็อกเป็น 4 และ 8 (ภาพที่ 28 และ 29 ตามลำดับ) Hybrid LZWGA เมื่อเปรียบเทียบกับ Hybrid GA จะเห็นว่า Hybrid LZWGA มี

ประสิทธิภาพในการทำงานในปัญหา RandomRoyalRoad ดีกว่ากับ Hybrid GA ที่ขนาดบล็อกล็อกเท่ากับ 4 อยู่ 10 เปอร์เซ็นต์ และที่ขนาดบล็อกล็อกเท่ากับ 8 อยู่ 23 เปอร์เซ็นต์



ภาพที่ 28 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA และ Hybrid GA ที่ขนาดบล็อกล็อกเท่ากับ 4



ภาพที่ 29 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad เปรียบเทียบระหว่าง Hybrid LZWGA และ Hybrid GA ที่ขนาดบล็อกล็อกเท่ากับ 8

วิจารณ์

1. วิจารณ์ผลการทดลองเปรียบเทียบประสิทธิภาพระหว่าง LZWGA และ GA ในปัญหา RandomMax และ RandomRoyalRoad

การเปรียบเทียบประสิทธิภาพของ LZWGA และ GA ในปัญหา RandomMax ที่ค่าความสุ่มเป็น 0 (ภาพที่ 15) ซึ่งเปรียบได้กับปัญหา OneMax นั้นเอง จากภาพแสดงให้เห็นว่า LZWGA ค้นพบคำตอบได้เร็วกว่า GA มาก โดยสามารถพบคำตอบตั้งแต่ประชากรในรุ่นแรก ๆ ของการวิวัฒนาการ ในขณะที่ GA จะพบคำตอบที่ประมาณรุ่นที่ 60 จากกราฟจึงได้นำเสนอถึงแค่รุ่นที่ 60 เนื่องจากหลังจากนั้นกราฟที่ได้ของ LZWGA และ GA จะเป็นเส้นตรงคงที่ แต่เมื่อเปลี่ยนค่าความสุ่มเป็น 1 LZWGA จะหาคำตอบไม่พบ (ภาพที่ 16) ในขณะที่ประสิทธิภาพของ GA ไม่เปลี่ยนแปลงถึงแม้ค่าความสุ่มจะเปลี่ยนจาก 0 เป็น 1

สำหรับปัญหา RandomRoyalRoad ที่ค่าความสุ่มเป็น 0 และขนาดบล็อกเป็น 4 และ 8 (ภาพที่ 17 และ 19) LZWGA สามารถหาคำตอบได้เร็วกว่า GA แต่เมื่อเปลี่ยนค่าความสุ่มเป็น 1 (ภาพที่ 18 และ 20) ทั้ง LZWGA และ GA ไม่สามารถหาคำตอบพบ และเนื่องจากในการทดลองได้จำกัดจำนวนรุ่นไว้ที่ 200 รุ่น ทำให้เห็นว่าภายใน 200 รุ่น LZWGA ได้คำตอบที่มีค่าความเหมาะสมน้อยกว่า GA ในขณะที่ประสิทธิภาพของ GA ไม่เปลี่ยนแปลงถึงแม้ค่าความสุ่มจะเปลี่ยนจาก 0 เป็น 1

2. วิจารณ์ผลการทดลองเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA และ LZWGA ในปัญหา RandomMax และ RandomRoyalRoad

จากภาพที่ 21, 22 และ 23 แสดงให้เห็นว่าเมื่อเพิ่มเทคนิคการค้นหาเฉพาะที่ให้ทำงานร่วมกับ LZWGA นั่นคือ Hybrid LZWGA ทำให้ประสิทธิภาพในการทำงานกับปัญหา RandomMax ดิซัน 22 เฟอร์เซนต์ และ RandomRoyalRoad ที่ขนาดบล็อกเท่ากับ 4 และ 8 ดิซัน 5 เฟอร์เซนต์และ 7 เฟอร์เซนต์ตามลำดับ แต่ก็ยังไม่สามารถหาคำตอบได้ เนื่องจาก LZWGA จะทำงานได้ดีในปัญหาที่คำตอบมีแบบแผน (เป็น 1 หมดทุกบิต) และอาจขึ้นอยู่กับเทคนิคการค้นหาเฉพาะที่ที่ใช้ยังไม่เหมาะสม

3. วิจารณ์ผลการทดลองเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA ทั้ง 4 แบบ ในปัญหา RandomMax และ RandomRoyalRoad

จากการทดลองก่อนหน้านี้ที่ใช้การค้นหาเฉพาะที่เข้ามาทำงานร่วมกับ LZWGA โดยทำการค้นหาเฉพาะที่กับโครโมโซมตัวที่ดีที่สุดขณะบิบบอดอยู่ในแต่ละรุ่น จึงได้เพิ่มกรณีในการทำการค้นหาเฉพาะที่อีก 3 แบบ ได้แก่

1. Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับทุกโครโมโซม ขณะบิบบอดอยู่
2. Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับทุกโครโมโซม ขณะที่คลายแล้ว
3. Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับเฉพาะตัวที่ดีที่สุด ขณะที่คลายแล้ว

จากภาพที่ 24, 25 และ 26 แสดงการเปรียบเทียบประสิทธิภาพของ Hybrid LZWGA ทั้ง 4 แบบ แสดงให้เห็นว่า Hybrid LZWGA แบบที่ทำการค้นหาเฉพาะที่กับเฉพาะโครโมโซมตัวที่ดีที่สุดขณะบิบบอดอยู่ ให้ประสิทธิภาพต่ำสุด ส่วน Hybrid LZWGA แบบที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัวขณะคลายแล้ว ให้ประสิทธิภาพดีที่สุด โดยดีขึ้นจากของเดิมประมาณ 16 เปอร์เซ็นต์ในปัญหา RandomMax และ ดีขึ้น 50 และ 40 เปอร์เซ็นต์ในปัญหา RandomRoyalRoad ที่ขนาดบล็อกละ 4 และ 8 ตามลำดับ และ Hybrid LZWGA แบบที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัว จะมีประสิทธิภาพค่อนข้างดีกว่าแบบที่ค้นหาเฉพาะตัวที่ดีที่สุดเพียงตัวเดียว เนื่องจากบางครั้งโครโมโซมตัวที่ดีที่สุดเมื่อปรับปรุงแล้ว ค่าความเหมาะสมใหม่ที่จะเป็นไปได้คือ ค่าความเหมาะสมที่เท่ากับหรือมากกว่าของเดิม แต่ก็อาจจะไม่ใช่ค่าที่ดีที่สุดในรอบถัดไป ส่วนโครโมโซมตัวที่ค่าความเหมาะสมไม่มากเมื่อปรับปรุงแล้ว อาจจะทำให้ค่าความเหมาะสมเพิ่มสูงกว่าตัวที่ดีที่สุดก็เป็นไปได้

4. วิจารณ์ผลการทดลองเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA และ LZWGA ในปัญหา RandomMax และ RandomRoyalRoad

จากผลการทดลอง Hybrid LZWGA แบบที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัว ขณะที่คลายแล้ว ให้ประสิทธิภาพดีที่สุด ซึ่ง Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมขณะที่คลายแล้วเปรียบได้กับการนำ GA มาทำการค้นหาเฉพาะที่ ดังนั้นผู้วิจัยจึงทำการเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัว ขณะที่คลายแล้วกับ Hybrid GA ที่นำโครโมโซม GA ทุกตัวไปทำการค้นหาเฉพาะที่

จากผลการทดลอง แสดงให้เห็นว่า ในปัญหา RandomMax (ภาพที่ 27) Hybrid LZWGA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัวขณะที่คลายแล้ว ให้ประสิทธิภาพในการค้นหาคำตอบต่ำกว่า Hybrid GA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัว โดยประสิทธิภาพต่ำกว่าประมาณ 5 เปอร์เซ็นต์และ Hybrid LZWGA ยังใช้เวลาในการทำงานเพิ่มมากขึ้น โดยใช้เวลาในการทำงานถึง 50 ชั่วโมง ในขณะที่ Hybrid GA ใช้เวลาเพียง 32 นาที

ในปัญหา RandomRoyalRoad ที่ขนาดบล็อกเท่ากับ 4 (ภาพที่ 28) และขนาดบล็อกเท่ากับ 8 (ภาพที่ 29) Hybrid LZWGA มีประสิทธิภาพในการหาคำตอบได้ดีกว่า Hybrid GA โดยดีกว่าประมาณ 10 เปอร์เซ็นต์ และ 23 เปอร์เซ็นต์ตามลำดับ ถึงแม้ว่าประสิทธิภาพในการหาคำตอบจะดีกว่าแต่ Hybrid LZWGA มีจุดด้อยในเรื่องของเวลาในการหาคำตอบ เนื่องจากใช้เวลาถึง 48 และ 44 ชั่วโมง ที่ขนาดบล็อกเท่ากับ 4 และ 8 ตามลำดับ ในขณะที่ Hybrid GA ใช้เวลาในการหาคำตอบที่ 38 นาทีและ 42 นาที

4. วิจัยผลการทดลองโดยรวม

ผลการทดลองโดยรวมพบว่า Hybrid LZWGA ให้ผลการทดลองค่อนข้างดีกว่า GA ในแง่ของการพบคำตอบที่เหมาะสม แต่ด้อยกว่า GA ในแง่ของเวลา เนื่องจาก Hybrid LZWGA ต้องใช้เวลาเพิ่มขึ้นในส่วนของการทำการค้นหาเฉพาะที่และการบีบ-คลายโครโมโซม ซึ่งส่วนต่างของเวลาถือว่าต่างกันมาก โดยจุดนี้สามารถนำไปพัฒนาต่อได้

สรุปและข้อเสนอแนะ

สรุป

ผู้วิจัยใช้จำนวนรุ่นและค่าความเหมาะสมในการประมวลผลในการวัดประสิทธิภาพ โดยทดลองกับปัญหาใหม่ที่มีพื้นฐานมาจากปัญหาแบบเดิมแต่ใช้การสุ่มเพื่อหาคำตอบทำให้ได้คำตอบที่ไม่มีแบบแผน ปัญหานั้น คือ RandomMax และ RandomRoyalRoad

LZWGA ไม่สามารถแก้ปัญหาลักษณะสุ่มได้ และประสิทธิภาพในการหาคำตอบยังเทียบเท่า GA ไม่ได้ ผู้วิจัยจึงปรับปรุง LZWGA โดยรวมการค้นหาเฉพาะที่เข้าไป เรียกว่า Hybrid LZWGA และทดสอบกับปัญหา RandomMax และ RandomRoyalRoad ผลการทดลองพบว่า Hybrid LZWGA ที่ใช้การค้นหาเฉพาะที่ปรับปรุงโครโมโซมทุกตัวขณะที่คลายแล้ว มีประสิทธิภาพการทำงานเพิ่มขึ้นจากของเดิม และยังมีประสิทธิภาพดีกว่า GA ในการแก้ปัญหามาตรฐาน RandomRoyalRoad ที่ขนาดบล็อกละ 4 และ 8 แต่ในปัญหา RandomMax ประสิทธิภาพยังเทียบเท่า GA ไม่ได้

Hybrid LZWGA ที่ใช้การค้นหาเฉพาะที่ปรับปรุงโครโมโซมทุกตัวขณะที่คลายแล้ว เปรียบได้กับ GA ที่ทำการค้นหาเฉพาะที่ หรือ Hybrid GA ผู้วิจัยจึงได้ทดลองเปรียบเทียบประสิทธิภาพระหว่าง Hybrid LZWGA และ Hybrid GA ที่ทำการค้นหาเฉพาะที่กับโครโมโซมทุกตัว ผลการทดลองแสดงให้เห็นว่า ประสิทธิภาพของ Hybrid LZWGA ดีกว่า Hybrid GA ในปัญหา RandomRoyalRoad ที่ขนาดบล็อกละ 4 และ 8 แต่ในปัญหา RandomMax ประสิทธิภาพยังต่ำกว่า Hybrid GA

ถึงแม้ว่าประสิทธิภาพโดยรวมของ Hybrid LZWGA จะดีกว่า GA และ Hybrid GA ก็ตาม แต่ถ้ามองในแง่ของเวลาที่ใช้ในการวิวัฒนาการ Hybrid LZWGA ใช้เวลานานกว่า Hybrid GA มาก โดย Hybrid LZWGA ใช้เวลาเป็นหลักชั่วโมง ในขณะที่ Hybrid GA ใช้เวลาไม่เกินหนึ่งชั่วโมง

ข้อเสนอแนะ

แม้ว่าประสิทธิภาพโดยรวมของ Hybrid LZWGA จะดีขึ้น แต่ก็ยังมีจุดด้อยอยู่คือ ต้องใช้เวลาในการทำงานเพิ่มขึ้น เนื่องจากต้องทำงานในส่วนของอัลกอริทึม LZWGA แล้วยังต้องทำการค้นหาเฉพาะที่ด้วย ซึ่งเวลาที่เพิ่มขึ้นมานี้ส่วนใหญ่เกิดจากการทำการค้นหาเฉพาะที่ ดังนั้น ถ้าจะปรับปรุงประสิทธิภาพของ Hybrid LZWGA ให้ดีขึ้น จึงควรปรับปรุงที่อัลกอริทึมการค้นหาเฉพาะที่ให้ใช้เวลาในการทำงานน้อยลงและเลือกใช้อัลกอริทึมการค้นหาเฉพาะที่ที่เหมาะสมกับปัญหาที่นำไปใช้

เอกสารและสิ่งอ้างอิง

นริศ กุณาศล, วรเศรษฐ สุวรรณิก และ ประภาส จงสถิตย์วัฒนา. 2548. การเข้ารหัสแบบ LZW ใน ขั้นตอนวิธีเชิงพันธุกรรม, น. 861-864. ใน **Proceedings of Electrical Engineering Conference (EECON-28)**.

วรเศรษฐ สุวรรณิก, นริศ กุณาศล และ ประภาส จงสถิตย์วัฒนา. 2549. ขั้นตอนวิธีเชิงพันธุกรรม แบบบีบอัด, น. 203-211. ใน **Proceedings of Northeastern Computer Science and Engineering Conference (NECSEC)**.

สาวิตรี น้าลมัย, วรเศรษฐ สุวรรณิก. 2550. ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด, น. 653-656. ใน **Proceedings of Electrical Engineering Conference (EECON-XXX)**.

สุภัคกานดา ชมภูมิ่ง, ประพล อธิพิงษ์ และ ภูพงษ์ พงษ์เจริญ. 2550. การประยุกต์ใช้เทคนิคซัพเฟิล ฟรอกลิปปิงเพื่อเพิ่มประสิทธิภาพการทำงานของมีมติกอัลกอริทึม, น. 162-174. ใน การประชุมวิชาการด้านการวิจัยดำเนินงานแห่งชาติ.

B. Freisleben and P. Merz. 1996. A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems, pp. 616-621. In **IEEE International Conference on Evolutionary Computation**.

Fengjie Wu. 2001. **A Framework for Memetic Algorithms**. M.S. Thesis, University of Auckland.

Melanie Mitchell. 1998. An Introduction to Genetic Algorithms (Complex Adaptive Systems). **MIT Press**. Cambridge, Massachusetts.

Moscato P. 2002. **Memetic Algorithms' Home Page**. From:

http://www.densis.fee.unicamp.br/~moscato/memetic_home.html, Accessed 9th October 2008.

Naris Kunasol, Worasait Suwannik and Prabhas Chongstitvatana. 2006. Solving One-Million-Bit Problems Using LZWGA, 32-36. **Proceedings of International Symposium on Communications and Information Technologies (ISCIT)**.

Nguyen Quoc Viet Hung, Ta Quang Binh and Duong Tuan Anh. 2005. A Memetic Algorithm for Timetabling. 289 – 294. *In* **Proceedings of 3rd Int. Conf. RIVF'05 Research Informatics Vietnam-Francophony**.

Stuart Russell and Peter Norvig. 1995. **Artificial Intelligence - A Modern Approach**. Prentice Hall. Englewood Cliffs, New Jersey.



ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด

Benchmark Problems for Compressed Genetic Algorithm

สาวตรี ฉ่ำมัย และ วรเศรษฐ สุวรรณิก

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

50 ถนนพหลโยธิน เขตจตุจักร กรุงเทพฯ 10900

โทร 0-2562-5555 โทรสาร 0-2942-8488 E-mail: g4964165@ku.ac.th, worasait.suwannik@gmail.com

บทคัดย่อ

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) เป็นอัลกอริทึมที่พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) LZWGA เข้ารหัสโครโมโซมในรูปแบบที่สามารถคลายได้ด้วยอัลกอริทึม Lempel-Ziv-Welch (LZW) ซึ่งช่วยลดขนาดของโครโมโซมและทำให้อัลกอริทึมสามารถแก้ปัญหาขนาดใหญ่ได้ อย่างไรก็ตามการวัดประสิทธิภาพได้กระทำกับปัญหาซึ่งคำตอบทุกบิตเป็น 1 หมด บทความนี้เสนอวิธีการวัดประสิทธิภาพของ LZWGA อีกสองวิธี คือ RandomMax และ RandomRoyalRoad ซึ่งมีพื้นฐานมาจากปัญหา OneMax และ RoyalRoad แต่ต่างกับปัญหาเดิมตรงที่คำตอบของปัญหาไม่ได้เป็นเลข 1 เหมือนกันทุกบิต การวัดค่าความเหมาะสมในโครโมโซมจะเปรียบเทียบกับคำตอบที่ถูกสุ่มขึ้นมา การทดลองแสดงให้เห็นว่า LZWGA สามารถแก้ปัญหา RandomRoyalRoad และ RandomMax ได้ดีกว่า GA เมื่อให้ค่าความสุ่มเป็น 0 แต่ที่ค่าความสุ่มเป็น 1 นั้น LZWGA จะดีกว่า GA

คำสำคัญ: ขั้นตอนวิธีเชิงพันธุกรรม, ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด ปัญหาทดสอบ

Abstract

LZW encoding in GA (LZWGA) is an algorithm developed from Genetic Algorithm (GA). LZWGA encodes chromosome in a format that can be decompressed using Lempel-Ziv-Welch (LZW) algorithm, which reduces the length of the chromosome and enables the algorithm to solve very large problems. However, the performance of the algorithm was evaluated using problems which every bit of solution is 1. This paper

proposes two benchmark problems: RandomMax and RandomRoyalRoad, which are based on OneMax and RoyalRoad respectively. The difference from the original problems are their solutions are not all 1's. The fitness is evaluated by comparing the chromosome with a random value. The experiment shows that LZWGA can solve RandomRoyalRoad and RandomMax better than GA when the randomness is 0 but its performance is worse than GA when the randomness is 1.

Keywords: RandomMax, RandomRoyalRoad, LZWGA, GA, OneMax, RoyalRoad

1. บทนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm - GA) เป็นวิธีการค้นหาที่เลียนแบบวิวัฒนาการของสิ่งมีชีวิต [1] การค้นหากระทำกับกลุ่มของโครโมโซม โครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมมีลักษณะเป็นสายบิต (binary string) เมื่อขนาดของโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหาก็จะมีขนาดใหญ่ขึ้นตามไปด้วย

LZWGA ใช้วิธีการเข้ารหัสโครโมโซมแบบบีบอัด ทำให้โครโมโซมมีขนาดเล็ก และการวิวัฒนาการกระทำกับโครโมโซมที่ถูกบีบอัด ในขณะที่ค่าความเหมาะสมของโครโมโซมจะวัดกับโครโมโซมที่ถูกคลาย [2] โครโมโซมที่ถูกบีบอัดเป็นอาร์เรย์ของจำนวนเต็ม เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำโครโมโซมนี้มาคลายจะได้โครโมโซมของ GA ปกติคือ 1010101

LZWGA สามารถแก้ปัญหาขนาดใหญ่มากถึง 1 ล้านบิต [3] ปัญหาดังกล่าวคือ OneMax, RoyalRoad, และ Trap โดยคำตอบของปัญหาดังกล่าวมีแบบแผนสูง (คำตอบเป็น 1 หมดทุกบิต) ซึ่งหากอัลกอริทึมมีประสิทธิภาพดีจริง

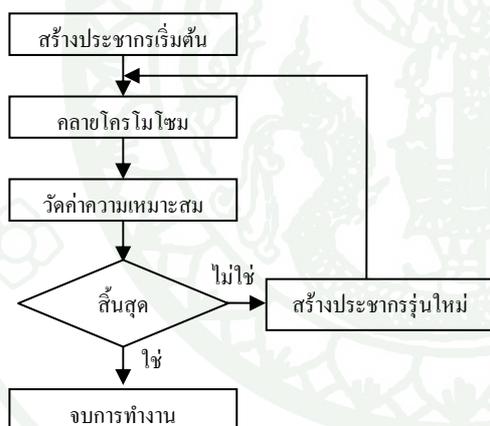
ไม่ว่าจะนำไปทดลองกับปัญหาใดประสิทธิภาพย่อมไม่ตกลงไป ดังนั้นเพื่อเป็นการพิสูจน์ประสิทธิภาพของ LZWGA จึงไม่สามารถใช้ปัญหามาตรฐานที่นิยมใช้กันในการวิจัยด้าน GA ได้

บทความนี้จึงนำเสนอปัญหาสำหรับทดสอบประสิทธิภาพของ LZWGA หรือขั้นตอนวิธีเชิงพันธุกรรมแบบบิพ-อัลโตๆ โดยปัญหาที่เสนอมีพื้นฐานจากปัญหามาตรฐาน แต่ใช้การสุ่มคำตอบเพื่อให้ได้คำตอบที่ไม่มีแบบแผน

2 LZWGA

2.1 ขั้นตอนการทำงาน

การทำงานของ LZWGA คล้ายกับการทำงานของ GA แต่ LZWGA จะมีกระบวนการกลายโครโมโซมเพิ่มเข้ามา โดยจะกลายโครโมโซมก่อนแล้วจึงวัดค่าความเหมาะสม ดังแสดงในรูปที่ 1



รูปที่ 1 กระบวนการทำงานของ LZWGA

2.2 การสร้างประชากรรุ่นแรก

โครโมโซมในรุ่นแรกของ LZWGA สร้างขึ้นโดยการสุ่มโดยให้อยู่ในเงื่อนไขที่สามารถนำไปกลายได้ เงื่อนไขดังกล่าวคือตำแหน่งที่ 1 ของโครโมโซมที่ถูกบิพอัลโตจะมีค่า 0 หรือ 1 ส่วนตำแหน่งถัดไปจะมีค่าได้ไม่เกินตำแหน่งนั้น เช่น โครโมโซมที่ถูกบิพตำแหน่งที่ 1 มีค่าอยู่ใน {0,1} โครโมโซมตำแหน่งที่ 2 มีค่าอยู่ใน {0,1,2} ซึ่งสามารถสรุปได้ตามสมการที่ 1

$$a_i = r \bmod (i+1), i \in \{1..n\} \tag{1}$$

โดย

a_i คือ ตำแหน่งที่ i ของโครโมโซมที่บิพอัลโตอยู่

r คือ เลขที่ถูกสุ่มขึ้นมา

n คือ ขนาดของโครโมโซม

2.3 การกลายและการวัดค่าความเหมาะสม

โครโมโซมที่ถูกบิพอัลโตของ LZWGA อยู่ในรูปแบบที่ยังบิพอัลโตอยู่ ดังนั้นก่อนนำโครโมโซมนี้ไปวัดค่าความเหมาะสมจึงต้องมีการกลายโครโมโซมก่อน ด้วยอัลกอริทึมการกลาย [2] ซึ่งหลังจากผ่าน อัลกอริทึมการกลายแล้วจะได้สายบิตเพื่อนำไปวัดค่าความเหมาะสมต่อไป

3 ปัญหา RandomMax

ปัญหา OneMax คือ ปัญหาการนับจำนวนเลขหนึ่งในโครโมโซม โดยค่าความเหมาะสมที่ได้ก็คือ จำนวนเลขหนึ่งทั้งหมดที่พบในโครโมโซม ดังนั้นโครโมโซมที่ดีที่สุดคือโครโมโซมที่ทุกบิตมีค่าเป็นหนึ่ง ปัญหา OneMax จำลองปัญหาการหาค่าเหมาะสมที่สุดที่ตัวแปรทุกตัวเป็นอิสระต่อกัน

ปัญหา RandomMax คือ ปัญหาการนับจำนวนบิตที่ตรงกับคำตอบในโครโมโซม โดยค่าความเหมาะสมที่ได้คือ จำนวนบิต m ตำแหน่งเดียวกับคำตอบมีค่าภายในตรงกัน ซึ่งคำตอบจะเกิดจากการสุ่ม เช่น

คำตอบ : 10100101

โครโมโซม : 10100100

ค่าความเหมาะสมที่ได้จะเป็นค่าของบิตในตำแหน่งเดียวกันที่มีค่าเท่ากันคือ 7 โดยอัลกอริทึมในการทำงานจะมี 2 ส่วนคือ ส่วนของการสุ่มคำตอบและส่วนของการวัดค่าความเหมาะสม ดังแสดงในรูปที่ 2

ในการสร้างคำตอบที่เกิดจากสุ่มนั้น จะมีการกำหนดค่าความสุ่ม ซึ่งในการทดลองจะมีสองค่าคือ 0 และ 1 โดยค่าความสุ่มเท่ากับ 0 คือไม่มีการสุ่มเกิดขึ้นเลย (คำตอบที่ได้เป็น 1 ทั้งหมด) และค่าความสุ่มเท่ากับ 1 คือทุกบิตของโครโมโซมจะเกิดจากการสุ่มหมด

```

CreateTargetChromosome()
begin
  for i=0 to chromosomeLength-1
    target[i]←random
  end for
end

int FitnessEvaluate(int[] chromosome)
begin
  fitness←0
  for i=0 to chromosomeLength-1
    if chromosome[i]=target[i]
      fitness++
    end if
  end for
  return fitness
end

```

รูปที่ 2 รหัสเทียมของ RandomMax กรณีที่ค่าความสุ่มเป็น 1

4 ปัญหา RandomRoyalRoad

ปัญหา RoyalRoad เป็นปัญหาที่จำลองปัญหาที่มีความสัมพันธ์ของบิตในโครโมโซม โดยคำตอบที่ต้องการคือค่าของบิตทุกตัวในบล็อกรวมค่าเท่ากับ 1 ในการวัดค่าความเหมาะสมหากบิตใดบิตหนึ่งในบล็อกรวมค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็นศูนย์ในทางกลับกัน หากทุกบิตในบล็อก มีค่าเป็น 1หมด จะได้ค่าความเหมาะสมเท่ากับจำนวนบิตในบล็อกนั้น

ปัญหา RandomRoyalRoad แตกต่างจาก RoyalRoad โดย ค่าค่าของบิตในบล็อกรวมค่ามีค่าเหมือนกับคำตอบซึ่งเกิดจากการสุ่ม ความเหมาะสมที่ได้เท่ากับจำนวนบิตในบล็อก แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่เหมือนคำตอบบล็อก นั้นจะได้ค่าความเหมาะสมเป็น 0ทันที เช่น

คำตอบ : 1110 1110 1110 1110

โครโมโซม : 1011 1110 1001 1110

ค่าความเหมาะสมที่ได้จะเท่ากับ 8 เนื่องจากในบล็อกรวมค่า 2 และ 4 โครโมโซมมีค่าเท่ากับคำตอบทุกตำแหน่งส่วนบล็อกรวมค่า 1 และ 3 จะไม่ได้คะแนนเนื่องจากมีบางบิตในโครโมโซมที่มีค่าไม่ตรงกับคำตอบ โดยอัลกอริทึมของ RandomRoyalRoad จะมีการทำงานหลัก 2 ส่วนคือ ส่วนของการสุ่มบล็อกรวมค่าและส่วนของการวัดค่าความเหมาะสม ดังแสดงในรูปที่ 3

```

CreateTargetChromosome()
begin
  for i=0 to blockSize-1
    block[i]←random
  end for
  m←0
  for j=0 to chromosomeLength-1
    for k=0 to blockSize-1
      target[m]←block[k]
      m++
    end for
    j←j+blockSize
  end for
end

int FitnessEvaluate(int[] chromosome)
begin
  count←0,fitness←0,k←0
  for i=0 to chromosomeLength-1
    for j=0 to blockSize-1
      if chromosome[k]=target[k]
        count++
        k++
      end if
    end for
    if count=blockSize
      fitness++
    end if
    i←i+blockSize
  end for
  return fitness*blockSize
end

```

รูปที่ 3 รหัสเทียมของ RandomRoyalRoad กรณีที่ค่าความสุ่มเป็น 1

5 การทดลอง

เราเปรียบเทียบประสิทธิภาพของ LZWGA และ GA โดยทดลองกับปัญหา RandomMax และ RandomRoyalRoad ขนาดหนึ่งพันบิต ค่าพารามิเตอร์ที่ใช้ในแต่ละอัลกอริทึมเป็นค่าที่ทำให้อัลกอริทึมได้ผลดีจากการทดลองเบื้องต้น ทั้งนี้เพราะว่าค่าพารามิเตอร์มีผลต่อคำตอบที่ได้ โดยค่าพารามิเตอร์ที่ได้แสดงในตารางที่ 1 และ 2

เนื่องจากอัลกอริทึมที่ใช้ในการทดลองนี้เป็นอัลกอริทึมที่มีการสุ่มตัวเลข เราจึงทำการทดลอง 10 ครั้งแล้วหาค่าเฉลี่ย โดยปัญหาที่ทดลองคือ

1. เปรียบเทียบประสิทธิภาพของ LZWGA กับ GA ในการแก้ปัญหา RandomMax ที่ค่าความสุ่มเป็น 0 และ 1

2. ประสิทธิภาพของ LZWGA กับ GA ในปัญหา RandomRoyalRoad ที่ค่าความสุ่มเป็น 0 และ 1 โดยกำหนดให้ขนาดบล็อกเป็น 4 และ 8

ตารางที่ 1 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomMax

พารามิเตอร์	LZWGA	GA
จำนวนประชากร	1000 ตัว	
ความยาวของโครโมโซมที่ถูกบีบอัด	168	-
ค่าความเหมาะสมของคำตอบ	1,000	
วิธีการคัดเลือก	แบบประลอง โครโมโซมที่ประลองกันมี 4 ตัว	
วิธีการไขว้เปลี่ยน	No crossover	Uniform
อัตราการไขว้เปลี่ยน	0%	80%
อัตราการกลายพันธุ์	1%	0%
เก็บโครโมโซมที่ดีที่สุดไว้	10 ตัว	
จำนวนรุ่นจำกัดไว้ที่	200 รุ่น	
ค่าความสุ่ม	0, 1	

ตารางที่ 2 พารามิเตอร์ที่ใช้ในการแก้ปัญหา

RandomRoyalRoad

พารามิเตอร์	LZWGA	GA
จำนวนประชากร	1000 ตัว	
ความยาวของโครโมโซมที่ถูกบีบอัด	168	-
ค่าความเหมาะสมของคำตอบ	1,000	
วิธีการคัดเลือก	แบบประลอง โครโมโซมที่ประลองกันมี 4 ตัว	
อัตราการไขว้เปลี่ยน	80%	80%
วิธีการไขว้เปลี่ยน	Single point	Uniform
อัตราการกลายพันธุ์	1%	0%
เก็บโครโมโซมที่ดีที่สุดไว้	10 ตัว	
จำนวนรุ่นจำกัดไว้ที่	200 รุ่น	
ค่าความสุ่ม	0, 1	

6 ผลการทดลอง

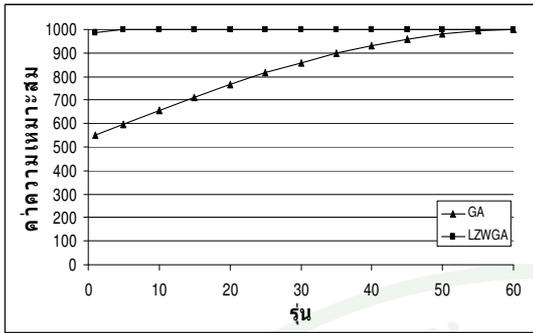
การเปรียบเทียบประสิทธิภาพของ LZWGA และ GA ในปัญหา RandomMax ที่ค่าความสุ่มเป็น 0 ซึ่งก็คือปัญหา OneMax นั้นเอง (รูปที่ 4 (ก)) แสดงให้เห็นว่า LZWGA ค้นพบคำตอบได้เร็วกว่า GA มาก ซึ่งสอดคล้องกับงานวิจัย [3] โดยสามารถพบคำตอบตั้งแต่ประชากรในรุ่นแรก ๆ ของการวิวัฒนาการ ในกรณีที่ GA จะพบคำตอบที่ประมาณรุ่นที่ 60 จากกราฟจึงได้นำเสนอถึงแก่รุ่นที่ 60 เนื่องจากหลังจากนั้นกราฟที่ได้ของ LZWGA และ GA จะเป็นเส้นตรงลงที่ แต่เมื่อเปลี่ยนค่าความสุ่มเป็น 1 LZWGA

จะหาคำตอบไม่พบ (รูป 4 (ข)) ในขณะที่ประสิทธิภาพของ GA ไม่เปลี่ยนแปลงถึงแม้ค่าความสุ่มจะเปลี่ยนจาก 0 เป็น 1

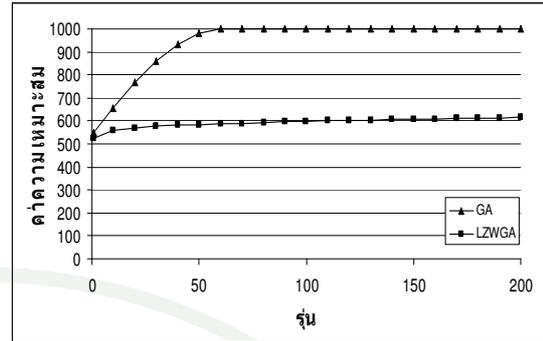
สำหรับปัญหา RandomRoyalRoad ที่ค่าความสุ่มเป็น 0 และขนาดบล็อกเป็น 4 และ 8 (รูปที่ 5 (ก) และ 6 (ก)) LZWGA สามารถหาคำตอบได้เร็วกว่า GA แต่เมื่อเปลี่ยนค่าความสุ่มเป็น 1 (รูปที่ 5 (ข) และ 6 (ข)) ทั้ง LZWGA และ GA ไม่สามารถหาคำตอบพบ และเนื่องจากในการทดลองได้จำกัดจำนวนรุ่นไว้ที่ 200 รุ่น ทำให้เห็นว่าภายใน 200 รุ่น LZWGA ได้คำตอบที่มีค่าความเหมาะสมน้อยกว่า GA ในขณะที่ประสิทธิภาพของ GA ไม่เปลี่ยนแปลงถึงแม้ค่าความสุ่มจะเปลี่ยนจาก 0 เป็น 1

7 สรุป

จากการทดลองกับปัญหา RandomMax และ RandomRoyalRoad ประสิทธิภาพของ GA ไม่เปลี่ยนแปลงไม่ว่าจะให้ค่าความสุ่มเป็น 0 หรือ 1 ก็ตาม ในขณะที่ LZWGA จะสามารถค้นหาคำตอบได้เร็วกว่า GA ในกรณีที่ค่าความสุ่มเป็น 0 แต่ในปัญหาที่ค่าความสุ่มเป็น 1 LZWGA ไม่สามารถค้นพบคำตอบที่ต้องการ โดยอาจเนื่องมาจากรูปแบบของค่าในโครโมโซมไม่มีแบบแผน เพราะค่าทุกบิตในโครโมโซมเกิดจากการสุ่ม ซึ่งในตอนนี้ผู้วิจัยจะศึกษาวิธีการแก้ปัญหาและปรับปรุงประสิทธิภาพต่อไปในงานวิจัย ภายหน้า

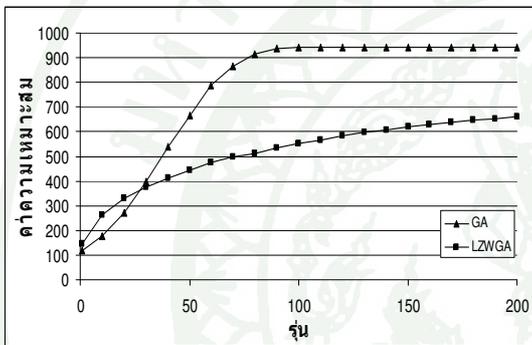


(ก)

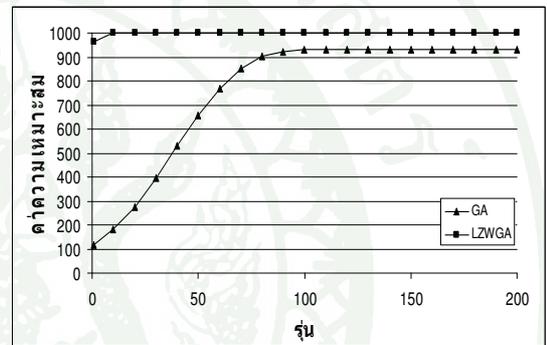


(ข)

รูปที่ 4 (ก) และ (ข) เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax ที่ค่าความสุ่มเท่ากับ 0 และ 1 ตามลำดับ

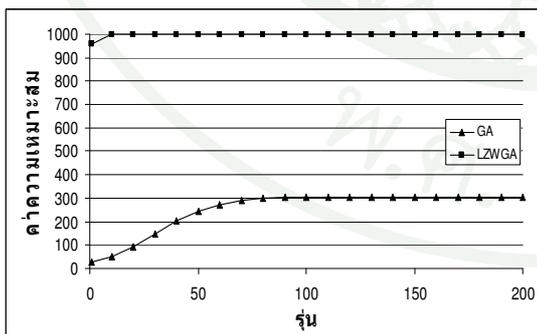


(ก)

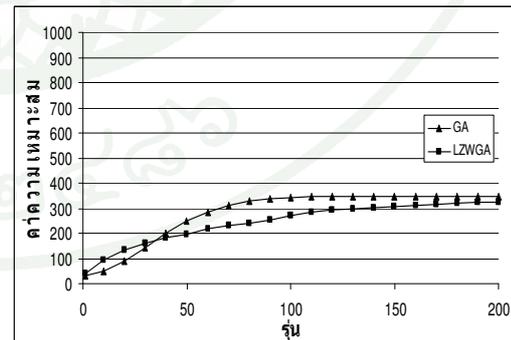


(ข)

รูปที่ 5 (ก) และ (ข) เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad ที่ค่าความสุ่มเท่ากับ 0 และ 1 และขนาดบล็อกร่วมเท่ากับ 4



(ก)



(ข)

รูปที่ 6 (ก) และ (ข) เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad ที่ค่าความสุ่มเท่ากับ 0 และ 1 และขนาดบล็อกร่วมเท่ากับ 8

เอกสารอ้างอิง

- [1] Melanie Mitchell, "An Introduction to Genetic Algorithms (Complex Adaptive Systems)," MIT Press, Cambridge, Massachusetts. ISBN: 0-262-63185-7, 1998.
- [2] นริศ กุณาศล, วรเศรษฐ สุวรรณิก และ ประภาส จงสถิตย์วัฒนา, "การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม," Proceedings of Electrical Engineering Conference (EECON-28), หน้า 861-864, 2548.
- [3] Naris Kunasol, Worasait Suwannik, Prabhas Chongstitvatana, "Solving One-Million-Bit Problems Using LZWGA," Proceedings of International Symposium on Communications and Information Technologies (ISCIT), 2006.



นางสาวสวิตรี ฉ่ำลมัย

สำเร็จการศึกษาระดับปริญญาตรีจาก สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ภาควิชาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์

ขณะนี้กำลังศึกษาต่อในระดับปริญญาโทภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์ งานวิจัยที่สนใจได้แก่ ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด

ขั้นตอนวิธีเชิงพันธุกรรม LZWGA แบบผสม

Hybrid LZWGA

สาวิตรี ชำลรัมย์¹ และ วรเศรษฐ์ สุวรรณิก¹

Sawitree Chumlamai¹ and Worasait Suwannik¹

บทคัดย่อ

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) เป็นอัลกอริทึมที่พัฒนา
มาจากขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) LZWGA เข้ารหัสโครโมโซมในรูปแบบที่สามารถ
คลายได้ด้วยอัลกอริทึม Lempel-Ziv-Welch (LZW) ซึ่งช่วยลดขนาดของโครโมโซมและทำให้อัลกอริทึม
สามารถแก้ปัญหาขนาดใหญ่มากได้ อย่างไรก็ตามการวัดประสิทธิภาพได้กระทำกับปัญหาซึ่งคำตอบทุกบิต
เป็น 1 ทั้งหมด ในงานวิจัยก่อนหน้านี้ ผู้วิจัยได้เคยนำเสนอวิธีการวัดประสิทธิภาพของ LZWGA คือ
RandomMax และ RandomRoyalRoad ที่มีพื้นฐานมาจากปัญหามาตรฐาน คือ OneMax และ
RoyalRoad แต่ต่างกันตรงที่คำตอบของปัญหาไม่ได้เป็นเลข 1 ทั้งหมดทุกบิต การวัดค่าความเหมาะสมใน
โครโมโซมจะเปรียบเทียบกับคำตอบที่ถูกสุ่มขึ้นมา การทดลองแสดงให้เห็นว่าประสิทธิภาพในการ
แก้ปัญหาแบบสุ่มของ LZWGA ไม่ดีนัก บทความนี้จึงได้นำเสนอการปรับปรุงประสิทธิภาพของ LZWGA
ในการแก้ปัญหาแบบสุ่ม โดยใช้เทคนิคการค้นหาเฉพาะที่ (Local Search) เข้ามาพร้อมด้วย เรียกว่า
Memetic Algorithm โดยแสดงให้เห็นว่าประสิทธิภาพของ LZWGA ที่ผสมการค้นหาเฉพาะที่ในการ
แก้ปัญหาแบบสุ่มดีขึ้น

ABSTRACT

LZW encoding in GA (LZWGA) is an algorithm developed from Genetic Algorithm (GA). LZWGA encodes chromosome in a format that can be decompressed using Lempel-Ziv-Welch (LZW) algorithm, which reduces the length of the chromosome and enables the algorithm to solve very large problems. However, the performance of the algorithm was evaluated using problems which every bit of solution is 1. In previous work we proposed two benchmark problems: RandomMax and RandomRoyalRoad based on standard problems -- OneMax and RoyalRoad. The difference from the original problems are their solutions are not all 1's. The fitness is evaluated by comparing the chromosome with a random value. The experiment shows that the efficiency of LZWGA in RandomMax and RandomRoyalRoad are unsatisfied. This paper proposes LZWGA combined with local search algorithm called

Memetic Algorithm to solve those random problems. From the experiment shows that the efficiency of it is better than LZWGA only.

Keywords: Hybrid LZWGA, RandomMax, RandomRoyalRoad, LZWGA, Memetic Algorithms.



S.Chumlamai: g4964165@ku.ac.th

¹ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
Department of Science, Faculty of Science, Kasetsart University.

คำนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm – GA) เป็นวิธีการค้นหาที่เลียนแบบวิวัฒนาการของสิ่งมีชีวิต (Mitchell, 1998) ถูกคิดค้นโดย John Holland โดยนำกระบวนการเปลี่ยนแปลงของสิ่งมีชีวิตตามธรรมชาติมาใช้ร่วมกับการเขียนโปรแกรม เมื่อขนาดของโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหา (search space) ก็จะมีขนาดใหญ่ขึ้นตามไปด้วย

LZWGA ใช้วิธีการเข้ารหัสโครโมโซมแบบบิต ทำให้โครโมโซมมีขนาดเล็กลง และการวิวัฒนาการกระทำกับโครโมโซมที่ถูกบิตบิต ในขณะที่ค่าความเหมาะสมของโครโมโซมจะวัดกับโครโมโซมที่ถูกคลายด้วยอัลกอริทึมการคลายของ LZW (นริศ, 2548) โครโมโซมที่ถูกบิตบิตเป็นอาร์เรย์ของจำนวนเต็ม โดย LZWGA สามารถแก้ปัญหามิติสูงมากถึง 1 ล้านบิต (นริศ, 2548) ปัญหาดังกล่าวคือ OneMax, RoyalRoad และ Trap โดยคำตอบของปัญหาดังกล่าวมีแบบแผนสูง (คำตอบเป็น 1 หมดทุกบิต)

ดังนั้นเพื่อเป็นการพิสูจน์ประสิทธิภาพของ LZWGA จึงไม่สามารถใช้ปัญหามาตรฐานที่นิยมใช้กันในการวิจัยด้าน GA ได้ จึงได้มีการนำเสนอปัญหาใหม่ สำหรับทดสอบประสิทธิภาพของ LZWGA โดยปัญหาที่นำเสนอมีพื้นฐานจากปัญหามาตรฐาน แต่ใช้การสุ่มคำตอบเพื่อให้ได้คำตอบที่ไม่มีแบบแผน (สรวิตร์, 2550) โดยได้ตั้งสมมติฐานภายใต้งานวิจัยของ LZWGA ที่สามารถทำงานได้เร็วกว่า GA ทั่วไป ถ้า LZWGA สามารถทำงานได้เร็วกว่า GA แล้ว ถ้านำ LZWGA ไปแก้ปัญหาก็ GA เคยแก้ได้ น่าจะทำให้การทำงานเร็วขึ้นกว่าเดิม แต่ผลการทดลองไม่เป็นไปตามที่ตั้งสมมติฐานที่ตั้งไว้ ซึ่งผลการทดลองทำให้เห็นว่า LZWGA มีประสิทธิภาพในการทำงานด้วยกว่า GA ในปัญหาแบบสุ่ม

Memetic Algorithm (MA) เป็นอัลกอริทึมแบบวิวัฒนาการ (evolutionary algorithm - EA) ที่รวมเทคนิคการค้นหาเฉพาะที่ (local search algorithm) เพื่อปรับปรุงประชากรแต่ละตัวให้ดีขึ้น (Fengjie, 2001) โดยวิธีการ Memetic Algorithm เป็นวิธีการที่ใช้ในการแก้ปัญหาค่าเหมาะสมที่สุด ซึ่งบางครั้งก็เรียกว่า Hybrid Genetic Algorithms (Moscato, 2002)

ได้มีงานวิจัยที่ทำการปรับปรุงขั้นตอนวิธีเชิงพันธุกรรม โดยรวมเทคนิคการค้นหาเฉพาะที่ (local search) ทดสอบกับ Traveling Saleman Problem (Freisleben, 1996) ซึ่งคำตอบที่แปลงให้อยู่ในรูปแบบของโครโมโซมจะอยู่ในรูปการเรียงแบบสับเปลี่ยนของตัวเลข (combinatorial optimization) โดยแสดงให้เห็นว่าประสิทธิภาพในการทำงานเพิ่มขึ้น นอกจากนี้ Nguyen Quoc Viet Hung (2005) ได้นำ Memetic Algorithm มาใช้ในการจัดตารางสอบของมหาวิทยาลัย โดยผลที่ได้แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมผสมกับเทคนิคการค้นหาเฉพาะที่สามารถแก้ปัญหการจัดตารางสอบแบบหลายจุดประสงค์ที่มีความยุ่งยากได้ ยังมีการนำ Memetic Algorithm ไปใช้ในการแก้ปัญหสมการคณิตศาสตร์ 2 สมการที่มีผลเฉลยแบบต่อเนื่อง (สุภัคกานดาและคณะ, 2550) โดยใช้เทคนิคการค้นหาเฉพาะที่ที่เรียกว่า ชัฟเฟิลฟรอกลิปปีง (shuffled frog leaping) และได้ข้อสรุปว่าการใช้เทคนิคชัฟเฟิล-

พหุคูณปิ้งใน Memetic Algorithm สามารถเพิ่มประสิทธิภาพของการประมาณค่าคำตอบที่ดีที่สุดได้ แต่ต้องใช้เวลาในการค้นหานานขึ้นกว่าเดิมเล็กน้อย

งานวิจัยหลายงานได้ทำการปรับปรุงขั้นตอนวิธีเชิงพันธุกรรมโดยให้ทำงานร่วมกับการค้นหาเฉพาะที่ และแสดงให้เห็นว่าประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมในการแก้ปัญหาที่มีประสิทธิภาพดีขึ้น ผู้วิจัยจึงได้พัฒนาอัลกอริทึม LZWGA โดยได้รวมเทคนิคการค้นหาเฉพาะที่ เพื่อเพิ่มประสิทธิภาพในการแก้ปัญหาแบบกลุ่ม เรียกว่า Hybrid LZWGA

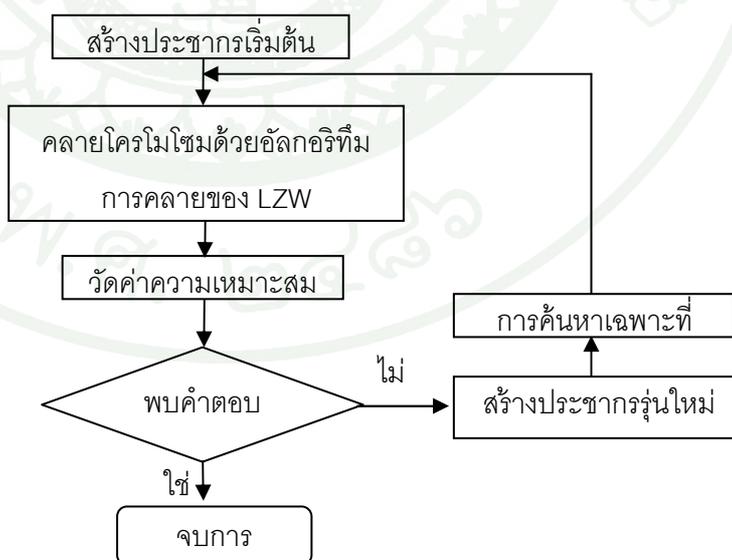
อุปกรณ์และวิธีการ

อุปกรณ์

ในการทดลอง ใช้เครื่องคอมพิวเตอร์ส่วนบุคคล Intel Core 2 Duo 1.6 GHz, หน่วยความจำ 1 GB. ระบบปฏิบัติการที่ใช้คือ Microsoft Window Vista พัฒนาผ่านโปรแกรม NetBeans IDE 6.1 และ Java Development Toolkit: JDK1.6

วิธีการ

งานวิจัยแบบผสมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดที่เข้ารหัสด้วยอัลกอริทึม LZW (Hybrid LZWGA) นี้ ได้นำอัลกอริทึมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดแบบ LZW มาใช้ร่วมกับวิธีการค้นหาเฉพาะที่ โดยจะนำการค้นหาเฉพาะที่มาใช้หลังการทำกระบวนการทางพันธุกรรม (genetic operator) แล้ว เพื่อหวังจะเพิ่มประสิทธิภาพของการทำงานให้ดีขึ้น โดยมีขั้นตอนการทำงาน ดังภาพที่ 1



ภาพที่ 1 แสดงขั้นตอนการทำงานของ Hybrid LZWGA

การสร้างประชากรเริ่มต้น

โครโมโซมในรุ่นแรกของ LZWGA สร้างขึ้นโดยการสุ่มโดยให้อยู่ในเงื่อนไขที่สามารถนำไปคลายได้ เงื่อนไขดังกล่าวคือ ตำแหน่งที่ 1 ของโครโมโซมที่ถูกบีบอัดจะมีค่า 0 หรือ 1 ส่วนตำแหน่งถัดไปจะมีค่าได้ไม่เกินตำแหน่งนั้น เช่น โครโมโซมที่ถูกบีบตำแหน่งที่ 1 มีค่าอยู่ใน $\{0,1\}$ โครโมโซมตำแหน่งที่ 2 มีค่าอยู่ใน $\{0,1,2\}$ ซึ่งสามารถสรุปได้ตามสมการที่ 1

$$a_i = r \bmod (i+1), i \in \{1..n\} \quad (1)$$

โดย

a_i คือ ตำแหน่งที่ i ของโครโมโซมที่บีบอัดอยู่

r คือ เลขที่ถูกสุ่มขึ้นมา

n คือ ขนาดของโครโมโซม

เมื่อได้โครโมโซมรุ่นแรกแล้ว จากนั้นก่อนนำโครโมโซมแต่ละตัวไปวัดค่าความเหมาะสม ต้องนำโครโมโซมไปทำการคลายด้วยอัลกอริทึมการคลายของ LZW ก่อน

การคลายและการวัดค่าความเหมาะสม

โครโมโซมที่ถูกบีบอัดของ LZWGA อยู่ในรูปแบบที่ยังบีบอัดอยู่ ดังนั้นก่อนนำโครโมโซมนี้ไปวัดค่าความเหมาะสมจึงต้องมีการคลายโครโมโซมก่อนด้วยอัลกอริทึมการคลาย ซึ่งหลังจากผ่านอัลกอริทึมการคลายแล้วจะได้สายบิตเพื่อนำไปวัดค่าความเหมาะสมต่อไป

การค้นหาเฉพาะที่

เทคนิคการค้นหาเฉพาะที่ ต่างจากขั้นตอนวิธีเชิงพันธุกรรมที่เป็นการค้นแบบครอบคลุม (global search) ซึ่งจะทำการค้นหาในปริภูมิการค้นหาโดยรวมได้ง่ายแต่จะมีผลให้การค้นหาคำตอบที่ดีที่สุดเป็นไปอย่างล่าช้า การค้นหาเฉพาะที่ จะทำการวนซ้ำ และเปลี่ยนค่าคำตอบของประชากรที่ดีอยู่แล้วให้ยิ่งดีขึ้นไปอีก เนื่องจากคำตอบที่ดีอยู่แล้วมักมีส่วนที่ไม่ดีรวมอยู่ด้วย

โดยในการทดลอง จะใช้การค้นหาเฉพาะที่เพื่อปรับปรุงประชากรตัวที่ดีที่สุด (best individual) เพียงตัวเดียว

Memetic Algorithms

ขั้นตอนวิธีนี้ได้แนวคิดมาจากทฤษฎี Memetics ของ Dawkins (1976) และจากนั้นได้นำเสนอโดย Moscato ในปี 1989 โดยเป็นการพัฒนาขั้นตอนวิธีเชิงพันธุกรรมให้ทำงานร่วมกับการค้นหาเฉพาะที่ โดยโครโมโซมในวิธีนี้มีติดกันจะถูกรเรียกว่า meme (Dawkins, 1976) ในการนำเทคนิคการค้นหาเฉพาะที่เข้ามาทำงานร่วมกับ GA เพื่อเพิ่มประสิทธิภาพในการหาคำตอบ โดยการทำงานอาจจัดกลุ่ม

เป็นกลุ่มย่อย แล้วใช้การค้นหาเฉพาะที่หาคำตอบที่ดีที่สุดในกลุ่ม (local maximum) จากนั้นใช้ GA เพื่อหาคำตอบที่ดีที่สุดของทั้งหมด (global maximum)

ปัญหาที่ใช้ในการทดลอง

ปัญหาที่ใช้ในการทดลองจะเป็นปัญหาแบบสุ่ม ได้แก่ ปัญหา RandomMax และ RandomRoyalRoad

ปัญหา RandomMax

ปัญหา OneMax คือ ปัญหาการนับจำนวนเลขหนึ่งในโครโมโซม โดยค่าความเหมาะสมที่ได้ก็คือ จำนวนเลขหนึ่งทั้งหมดที่พบในโครโมโซม ดังนั้นโครโมโซมที่ดีที่สุดคือโครโมโซมที่ทุกบิตมีค่าเป็นหนึ่งหมด

ปัญหา OneMax จำลองปัญหาการหาค่าเหมาะสมที่สุดที่ตัวแปรทุกตัวเป็นอิสระต่อกัน

ปัญหา RandomMax คือ ปัญหาการนับจำนวนบิตที่ตรงกับคำตอบในโครโมโซม โดยทำการสุ่มคำตอบเป็นสายบิตขึ้นมาก่อน จากนั้นจะนำไปวัดค่าความเหมาะสม โดยนำประชากรแต่ละตัวไปเปรียบเทียบกับคำตอบที่สุ่มขึ้นมาบิตต่อบิต ซึ่งค่าความเหมาะสมที่ได้คือ จำนวนบิต ณ ตำแหน่งเดียวกับคำตอบมีค่าภายในตรงกัน เช่น

คำตอบที่เกิดจากการสุ่ม : 10100101

โครโมโซม : 10100100

ค่าความเหมาะสมที่ได้จะเป็นค่าของบิตในตำแหน่งเดียวกันที่มีค่าเท่ากันคือ 7

การสร้างคำตอบที่เกิดจากสุ่มนั้น จะมีการกำหนดค่าความสุ่ม ซึ่งในการทดลองจะมีสองค่าคือ 0 และ 1 โดยค่าความสุ่มเท่ากับ 0 คือไม่มีการสุ่มเกิดขึ้นเลย (คำตอบที่ได้เป็น 1 หมด) และค่าความสุ่มเท่ากับ 1 คือทุกบิตของโครโมโซมจะเกิดจากการสุ่มหมด

ปัญหา RandomRoyalRoad

ปัญหา RoyalRoad เป็นปัญหาที่จำลองปัญหาที่มีความสัมพันธ์ของบิตในโครโมโซม โดยคำตอบที่ต้องการคือค่าของบิตทุกตัวในบล็อกมีค่าเท่ากับ 1 ในการวัดค่าความเหมาะสม หากบิตใดบิตหนึ่งในบล็อกมีค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็นศูนย์ ในทางกลับกัน หากทุกบิตในบล็อก มีค่าเป็น 1 หมด จะได้ค่าความเหมาะสมเท่ากับขนาดของบล็อกที่กำหนดไว้

ปัญหา RandomRoyalRoad แตกต่างจาก RoyalRoad โดยจะต้องมีการสุ่มคำตอบเป็นสายบิตขึ้นมาก่อน จากนั้นจะนำประชากรแต่ละตัวไปเปรียบเทียบกับคำตอบที่สุ่มขึ้นมาเพื่อวัดค่าความเหมาะสม ถ้าค่าของบิตในบล็อกนั้นทุกบิตมีค่าเหมือนกับคำตอบซึ่งเกิดจากการสุ่ม ความเหมาะสมที่ได้จะเท่ากับ

ขนาดของบล็อก แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่เหมือนกับคำตอบ บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันที เช่น

คำตอบที่เกิดจากการสุ่ม : 1110 1110 1110 1110

โครโมโซม : 1011 1110 1001 1110

จากตัวอย่าง ที่ขนาดบล็อกเท่ากับ 4 ค่าความเหมาะสมที่ได้จะเท่ากับ 8 เนื่องจากในบล็อกที่ 2 และ 4 โครโมโซมมีค่าเท่ากับคำตอบทุกตำแหน่ง ส่วนบล็อกที่ 1 และ 3 จะไม่ได้คะแนนเนื่องจากมีบางบิตในโครโมโซมที่มีค่าไม่ตรงกับคำตอบ

ผล

เปรียบเทียบประสิทธิภาพของ LZWGA และ Hybrid LZWGA โดยทดลองกับปัญหา RandomMax และ RandomRoyalRoad โครโมโซม LZWGA ขนาดสองร้อยบิต ค่าพารามิเตอร์ที่ใช้เป็นค่าที่อ้างอิงจากการทดลองเบื้องต้นของ LZWGA โดยจำกัดจำนวนรุ่นไว้ที่ 200 รุ่น จำนวนประชากรที่ใช้ 1,000 ตัว โดยค่าพารามิเตอร์ที่ใช้ แสดงในตารางที่ 1 และ 2

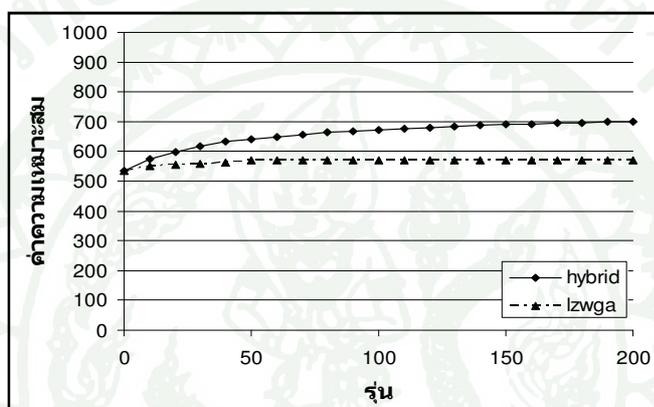
ตารางที่ 1 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomMax

พารามิเตอร์	ค่าที่ใช้
ค่าความเหมาะสมของคำตอบ	1,000
วิธีการคัดเลือก	แบบประลอง โครโมโซมที่ประลองกันมี 4 ตัว
วิธีการไขว้เปลี่ยน	ไม่ทำการไขว้เปลี่ยน
อัตราการกลายพันธุ์	1%
เก็บโครโมโซมที่ดีที่สุดไว้	10 ตัว

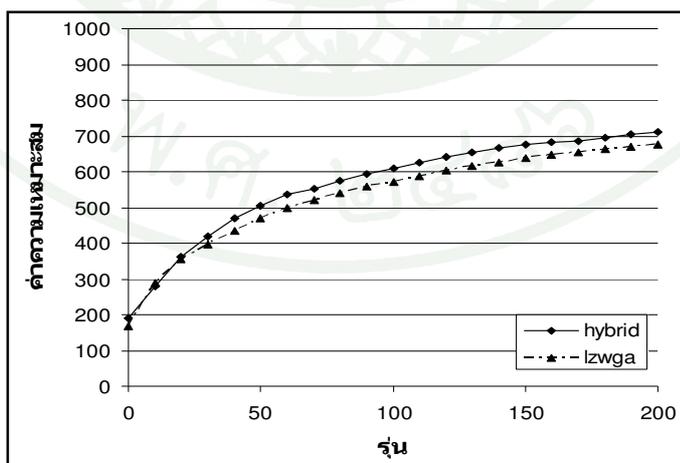
ตารางที่ 2 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad

พารามิเตอร์	ค่าที่ใช้
ค่าความเหมาะสมของคำตอบ	1,000
วิธีการคัดเลือก	แบบประลอง โครโมโซมที่ประลองกันมี 4 ตัว
อัตราการไขว้เปลี่ยน	80%
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบจุดเดียว
อัตราการกลายพันธุ์	1%
เก็บโครโมโซมที่ดีที่สุดไว้	10 ตัว

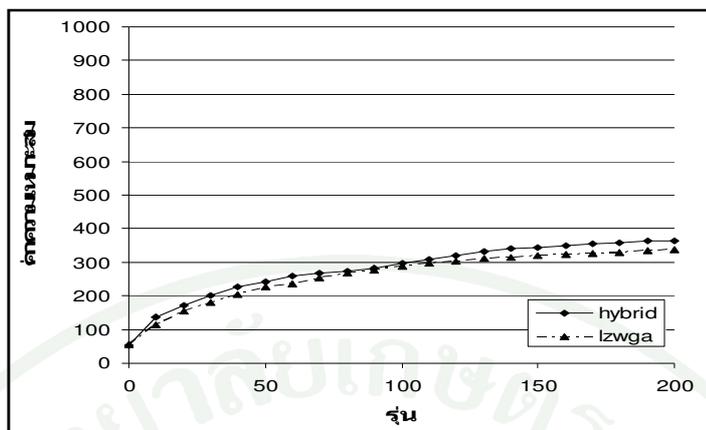
การเปรียบเทียบประสิทธิภาพของ LZWGA และ Hybrid LZWGA ค่าความเหมาะสมที่เป็นไปได้คือ ค่าความเหมาะสมที่เท่ากับ 1000 จากผลการทดลอง แสดงให้เห็นว่า Hybrid LZWGA มีประสิทธิภาพในการทำงานในปัญหา RandomMax (ภาพที่ 2) ดีขึ้นกว่า LZWGA โดยประสิทธิภาพในการทำงานดีขึ้นประมาณ 22 เปอร์เซ็นต์ สำหรับปัญหา RandomRoyalRoad ที่ขนาดบล็อกเป็น 4 และ 8 (รูปที่ 3 และ 4) Hybrid LZWGA สามารถหาคำตอบได้เร็วกว่า LZWGA เล็กน้อย โดยประสิทธิภาพดีขึ้นประมาณ 5 และ 7 เปอร์เซ็นต์ ตามลำดับ แต่อย่างไรก็ตาม ทั้ง Hybrid LZWGA และ LZWGA ก็ยังไม่สามารถหาคำตอบของปัญหาแบบสุ่มได้



ภาพที่ 2 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomMax



ภาพที่ 3 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad ขนาดบล็อกเท่ากับ 4



ภาพที่ 4 เส้นโค้งความเหมาะสมในการแก้ปัญหา RandomRoyalRoad ขนาดบล็อกลึกเท่ากับ 8

เนื่องจากการเปรียบเทียบผลการทำงานของ LZWGA และ Hybrid LZWGA โดยใช้กราฟเป็น ตัวตัดสินนั้น อาจยังไม่ชัดเจนถึงความแตกต่างที่พบ จึงได้นำวิธีการทางสถิติคือ การทดสอบแบบ T-Test มาใช้ในการวัดความแตกต่างในการทำงานระหว่าง LZWGA และ Hybrid LZWGA โดยใช้การทดสอบแบบ Paired Samples คือ กรณีที่กลุ่มตัวอย่างทั้งสองไม่อิสระต่อกัน เนื่องจากในการทดลอง ได้ใช้ค่าพารามิเตอร์ที่เหมือนกันทุกประการ(ตารางที่ 1 และ 2) ดังตารางที่ 3 โดยเปรียบเทียบระหว่าง LZWGA และ Hybrid LZWGA ในปัญหา RandomMax RandomRoyalRoad ที่ขนาดบล็อกลึกเป็น 4 และ 8 ตามลำดับ จากตาราง ที่ $t = 11.739$, $t = 11.953$ และ $t=9.877$ มีค่านัยสำคัญ (Sig.) เป็น .000 ซึ่งน้อยกว่า 0.05 ดังนั้น ปฏิเสธ H_0 นั่นคือ Hybrid LZWGA และ LZWGA มีความแตกต่างกันอย่างมีนัยสำคัญที่ระดับ 0.05

ตารางที่ 3 การทดสอบความแตกต่างของค่าเฉลี่ยกรณีทีกลุ่มตัวอย่างทั้งสองไม่อิสระต่อกัน จาก T-Test

HYBRID vs. LZWGA		Paired Differences					t	df	Sig. (2- tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
คู่มือ 1	Random Max	90.9333	35.49685	7.74605	74.7754	107.091 3	11.739	20	.000
คู่มือ 2	Random RoyalRoad block=4	31.6381	12.12924	2.64682	26.1169	37.1593	11.953	20	.000
คู่มือ 3	Random RoyalRoad block=8	18.7429	8.69601	1.89762	14.7845	22.7012	9.877	20	.000

วิจารณ์

จากรูปภาพที่ 2, 3 และ 4 แสดงให้เห็นว่าเมื่อเพิ่มเทคนิคการค้นหาเฉพาะที่ให้ทำงานร่วมกับ LZWGA นั่นคือ Hybrid LZWGA ทำให้ประสิทธิภาพในการทำงานกับปัญหา RandomMax และ RandomRoyalRoad ดีขึ้น แต่ก็ยังไม่สามารถหาคำตอบได้ เนื่องจาก LZWGA จะทำงานได้ดีในปัญหาที่คำตอบมีแบบแผน (เป็น 1 หมดทุกบิต) และอาจขึ้นอยู่กับเทคนิคการค้นหาเฉพาะที่ใช้ยังไม่เหมาะสม ซึ่งส่วนนี้ผู้วิจัยจะปรับปรุงต่อไปในงานวิจัยภายหน้า

สรุป

เปรียบเทียบประสิทธิภาพของ Hybrid LZWGA และ LZWGA กับปัญหา RandomMax และ RandomRoyalRoad แสดงให้เห็นว่า เทคนิคการค้นหาเฉพาะที่มีผลต่อประสิทธิภาพที่เพิ่มขึ้นของ Hybrid LZWGA คือ เมื่อเพิ่มเทคนิคการค้นหาเฉพาะที่เข้ามาทำงานร่วมกับ LZWGA ทำให้ประสิทธิภาพในการทำงานในปัญหา RandomMax ดีขึ้นประมาณ 22 เปอร์เซ็นต์ และประสิทธิภาพในการทำงานในปัญหา RandomRoyalRoad ดีขึ้นประมาณ 5 เปอร์เซ็นต์และ 7 เปอร์เซ็นต์ ที่ขนาดบิตถูกเท่ากับ 4 และ 8 ตามลำดับ

ประโยชน์ที่คาดว่าจะได้รับ

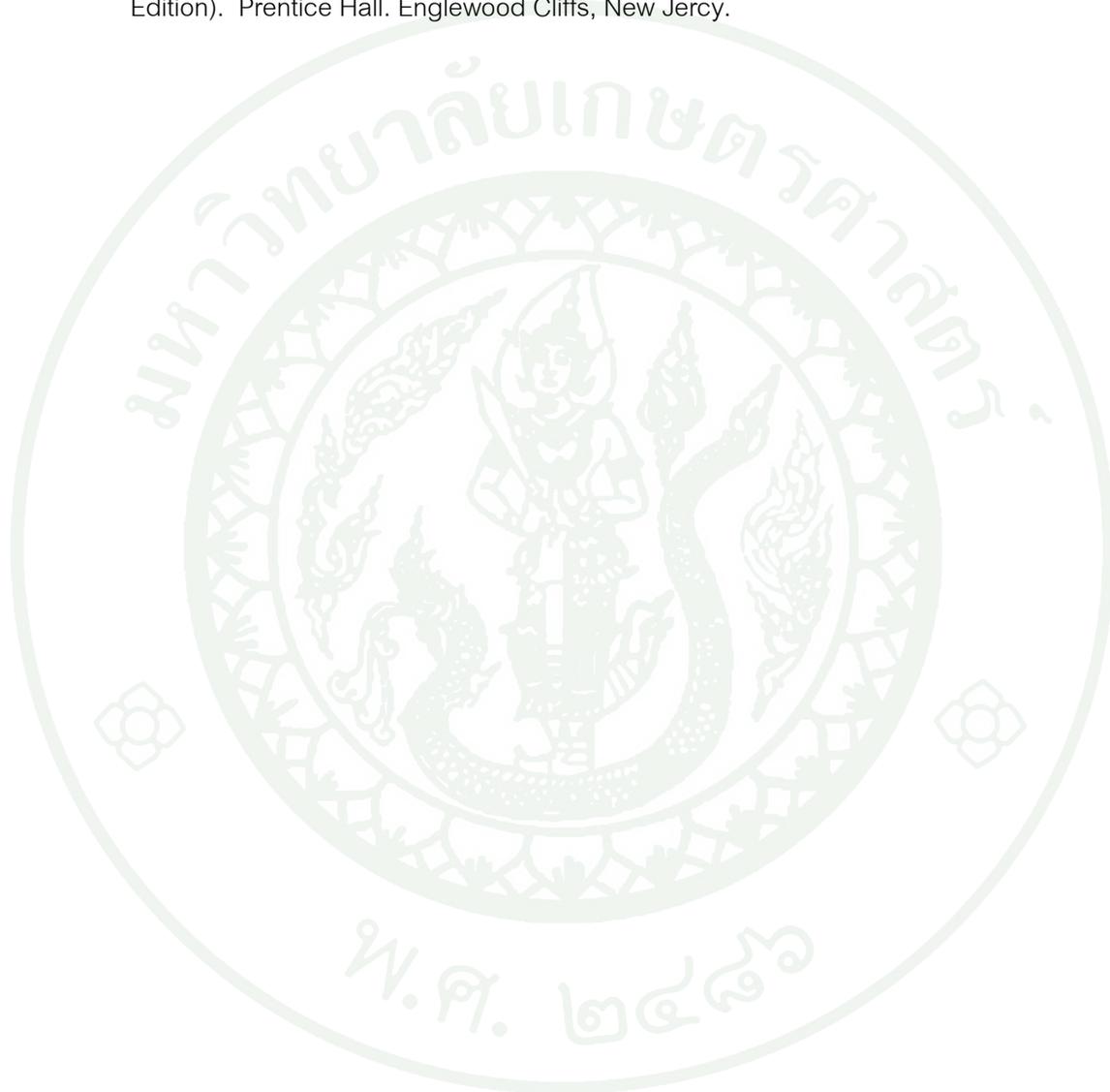
1. เพิ่มประสิทธิภาพของ LZWGA ให้สามารถแก้ปัญหาแบบสุ่มได้
2. สามารถนำเอาอัลกอริทึมจากการวิจัยไปประยุกต์ใช้สำหรับการแก้ปัญหาอื่น ๆ ต่อไป

เอกสารอ้างอิง

- นริศ กุณาศล, วรเศรษฐ สุวรรณิก และ ประภาส จงสถิตย์วัฒนา. 2548. การเข้ารหัสแบบ LZW ใน ขั้นตอนวิธีเชิงพันธุกรรม. Proceedings of Electrical Engineering Conference (EECON-28): 861-864.
- วรเศรษฐ สุวรรณิก, นริศ กุณาศล และ ประภาส จงสถิตย์วัฒนา. 2549. ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด, Proceedings of Northeastern Computer Science and Engineering Conference (NECSEC): 203-211.
- สาวิตรี ฉ่ำลมัย, วรเศรษฐ สุวรรณิก. 2550. ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด. Proceedings of Electrical Engineering Conference (EECON-XXX): 653-656.
- สุภัคกานดา ชมภูมิ่ง, ประพล อธิพิงษ์ และ ภูพงษ์ พงษ์เจริญ. 2550. การประยุกต์ใช้เทคนิคซัพเฟิลพรอกลิบปิงเพื่อเพิ่มประสิทธิภาพการทำงานของมีมิดิกอัลกอริทึม. การประชุมวิชาการด้านการวิจัยดำเนินงานแห่งชาติ: 162-174.
- B. Freisleben and P. Merz. 1996. A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems. IEEE International Conference on Evolutionary Computation: 616-621.
- Fengjie Wu. 2001. A Framewrok for Memetic Algorithms. M.S. Thesis, University of Auckland.
- Melanie Mitchell. 1998. An Introduction to Genetic Algorithms (Complex Adaptive Systems). MIT Press. Cambridge, Massachusetts.
- Moscato P. 2002. Memetic Algorithms' Home Page, accessed 9th Octobor 2008, from <http://www.densis.fee.unicamp.br/~moscato/memetic_home.html>.
- Naris Kunasol, Worasait Suwannik and Prabhas Chongstitvatana. 2006. Solving One-Million-Bit Problems Using LZWGA. Proceedings of International Symposium on Communications and Information Technologies (ISCIT): 32-36.

Nguyen Quoc Viet Hung, Ta Quang Binh and Duong Tuan Anh. 2005. A Memetic Algorithm for Timetabling. Proceedings of 3rd Int. Conf. RIVF'05 Research Informatics Vietnam-Francophony: 289 – 294.

Stuart Russell and Peter Norvig. 1995. Artificial Intelligence - A Modern Approach (2nd Edition). Prentice Hall. Englewood Cliffs, New Jersey.



ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	นางสาวสาวิตรี น้าลมัย
วัน เดือน ปี ที่เกิด	8 กันยายน 2526
สถานที่เกิด	จังหวัดนครราชสีมา
ประวัติการศึกษา	ระดับปริญญาตรี สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยี พระจอมเกล้าพระนครเหนือ
ตำแหน่งหน้าที่การงานปัจจุบัน	Technical Consultant
สถานที่ทำงานปัจจุบัน	บริษัท ยิบอินซอย จำกัด
ผลงานดีเด่นและรางวัลทางวิชาการ	-
ทุนการศึกษาที่ได้รับ	-