ใบรับรองวิทยานิพนธ์

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

ปริญญา

วิทยาการคอมพิวเตอร์ วิทยาการคอมพิวเตอร์

สาขา **ภาควิชา**

เรื่อง ตัวดำเนินการ Shift-and-Check สำหรับ LZWGA

A Shift-and-Check Operator for LZWGA

นามผู้วิจัย นางสาวอณิมา รอดเสียงล้ำ

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก .....

(ผู้ช่วยศาสตราจารย์วรเศรษฐ สุวรรณิก, วศ.ด.)

หัวหน้าภาควิชา .....

(ผู้ช่วยศาสตราจารย์ศิริกร จันทร์นวล, M.Sc.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

.....

(รองศาสตราจารย์กัญจนา ธีระกุล, D.Agr.)

คณบดีบัณฑิตวิทยาลัย

วันที่ ..... **เดือน** ..... **พ.ศ.** .....

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

วิทยานิพนธ์

เรื่อง

ตัวดำเนินการ Shift-and-Check สำหรับ LZWGA

A Shift-and-Check Operator for LZWGA

โดย

นางสาวอณิมา รอตเสียงล้ำ

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
เพื่อความสมบูรณ์แห่งปริญญาวิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)
พ.ศ. 2553

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

อณิมา รอดเสียงล้ำ 2553: ตัวดำเนินการ Shift-and-Check สำหรับ LZWGA ปริญญาวิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์) สาขาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก: ผู้ช่วยศาสตราจารย์ วรเศรษฐ สุวรรณิก, วศ.ด. 94 หน้า

งานวิจัยนี้มีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพการทำงานให้กับการเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) โดยผู้วิจัยเสนอตัวดำเนินการตัวใหม่ ชื่อว่า shift และ shift-and-check ตัวดำเนินการ shift เลื่อนตำแหน่งของโครโมโซมหลังตำแหน่งที่สุ่มได้ไปทางซ้ายหนึ่งตำแหน่ง ส่วนตัวดำเนินการ shift-and-check เลื่อนตำแหน่งของโครโมโซมแบบมีเงื่อนไข โดยหลีกเลี่ยงการทำให้ค่าของตำแหน่งที่เลื่อนไปเป็นศูนย์ นอกจากนี้ผู้วิจัยยังได้ทำการพัฒนาตัวดำเนินการ shift-and-check ให้เหมาะสมกับการนำไปใช้ในการแก้ปัญหาขนาดใหญ่ตั้งชื่อว่า ISC ตัวดำเนินการ ISC ดังกล่าวนี้จะเพิ่มการกำหนดอัตราการ shift ให้แตกต่างกันไปตามแต่ตำแหน่งของโครโมโซม ซึ่งตำแหน่งต้นๆจะมีโอกาสเกิดการ shift มากกว่าตำแหน่งท้ายๆ

การทดลองวัดประสิทธิภาพของตัวดำเนินการที่พัฒนาขึ้นทำกับ 4 ปัญหาทดสอบ คือ OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad โดยทดลองกับกลุ่มประชากรขนาดเล็ก ผลการทดลองพบว่าตัวดำเนินการ shift ช่วยเพิ่มประสิทธิภาพการทำงานของ LZWGA ได้ในทุกปัญหาทดสอบโดยทำให้โครโมโซมที่ได้มีค่าความเหมาะสมมากขึ้น ตัวดำเนินการ shift-and-check นั้นช่วยทำให้ค้นพบคำตอบได้เร็วขึ้น ส่วนตัวดำเนินการ ISC ที่ถูกปรับปรุงขึ้นมาเพื่อใช้กับปัญหาขนาดใหญ่นั้นก็สามารถช่วยปรับปรุงประสิทธิภาพการทำงานของ LZWGA ได้ โดยเพิ่มค่าความเหมาะสมให้แก่โครโมโซมในทุกรุ่นของวิวัฒนาการเช่นกัน รวมถึงช่วยทำให้พบคำตอบได้เร็วขึ้นด้วย

_____ _____ ___/___/___

ลายมือชื่อนิสิต ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

Animar Rotsianglum 2010: A Shift-and-Check Operator for LZWGA.
Master of Science (Computer Science), Major Field: Computer Science, Department of Computer Science. Thesis Advisor: Assistant Professor Worasait Suwannik, Ph.D.
94 pages.

The objective of this research is to improve the performance of LZWGA. We propose new operators call “shift” and “shift-and-check”. The shift operator left-shifts numbers in LZWGA chromosome. A position to be shifted was randomly selected. The shift-and-check operator adds a conditional that shifted positions cannot be zero after shifting. In addition we propose an improved shift-and-check operator and called “ISC” for solving a large problem. The shifting probability of ISC depends on its position. The left position had a higher shift rate than the right position.

For the experiment, we compare the performance of LZWGA by using the chromosome fitness value in 4 test problems: OneMax, RandomMax, RoyalRoad, and RandomRoyalRoad. We use a small population size. The results show that the shift operator helped LZWGA to produce a better chromosome in every test problems. The shift-and-check operator helps LZWGA to find a solution quickly. The ISC improved the performance of LZWGA for solving large problems.

_____ _____ ___ / ___ / ___

Student’s signature Thesis Advisor’s signature

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

กิตติกรรมประกาศ

ผู้วิจัยขอกราบขอบพระคุณ ผศ.ดร.วรเศรษฐ สุวรรณิก ประธานกรรมการที่ปรึกษา ที่ได้ให้คำปรึกษาในด้านการเรียน การค้นคว้าวิจัย ชี้แนะแนวทางแก้ไขปัญหา และตรวจแก้ไขวิทยานิพนธ์จนกระทั่งเสร็จสมบูรณ์

ขอกราบขอบพระคุณ อ.ดร.อุษา สัมมาพันธ์ ที่ได้ให้คำปรึกษาทั้งด้านการเรียนและการใช้ภาษาอังกฤษตลอดการทำวิจัยและขอกราบขอบพระคุณคณาจารย์ทุกท่านสำหรับการถ่ายทอดความรู้อันเป็นประโยชน์อย่างยิ่งตลอดช่วงเวลาในการศึกษาเล่าเรียน

ขอกราบขอบพระคุณสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท.) ที่ได้มอบทุนการศึกษาระดับปริญญาโท

ขอขอบคุณ คุณสมรักษ์ นุ่มนาค ที่ช่วยให้คำแนะนำในการทำวิจัยและวิทยานิพนธ์ ขอขอบคุณคุณหทัยเทพ ศิริธนะและคุณธนพล ตันติศรีปรีชาที่ช่วยไขปัญหาในการเขียนโปรแกรมด้วยดีมาโดยตลอด

ขอขอบคุณ คุณชุลีพร สืบสินและ คุณวิสิทธิ์ บุญชุม สำหรับความช่วยเหลือและความเป็นมิตรที่ดีเสมอ

ขอขอบคุณ ภาควิชาวิทยาการคอมพิวเตอร์ ที่ได้สนับสนุนด้านอุปกรณ์ และสถานที่ตลอดระยะเวลาในการทำวิทยานิพนธ์

ด้วยความดีหรือประโยชน์อันใดเนื่องจากวิทยานิพนธ์เล่มนี้ ขอมอบแด่ คุณพ่อสมนึกและคุณแม่โสภา รอตเสียงล้ำ ที่ได้กรุณาอบรมและให้กำลังใจผู้วิจัยมาตลอดในทุกเรื่อง

อณิมา รอตเสียงล้ำ

เมษายน 2553

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สารบัญ

หน้า

สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(4)
คำอธิบายสัญลักษณ์และคำย่อ	(6)
คำนำ	1
วัตถุประสงค์	3
การตรวจเอกสาร	5
อุปกรณ์และวิธีการ	21
อุปกรณ์	21
วิธีการ	21
ผลและวิจารณ์	33
ผล	33
วิจารณ์	64
สรุปและข้อเสนอแนะ	67
สรุป	67
ข้อเสนอแนะ	68
เอกสารและสิ่งอ้างอิง	69
ภาคผนวก	72
ประวัติการศึกษาและการทำงาน	94

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สารบัญตาราง

ตารางที่		หน้า
1	ค่าความเหมาะสมที่ได้หลังการ shift-and-check ที่ตำแหน่งต่างๆของโครโมโซม	25
2	ค่าของ P_i ของโครโมโซม LZWGA 4 ถึง 7 ตำแหน่ง	26
3	พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax	27
4	พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomMax	28
5	พารามิเตอร์ที่ใช้ในการแก้ปัญหา RoyalRoad	28
6	พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad	29
7	พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax และ RoyalRoad เพื่อเทียบความยาว	29
8	พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax และ RoyalRoad เพื่อเทียบจำนวนรุ่น	30
9	พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax และ RoyalRoad	31
10	พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad	31
11	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆใน ปัญหา OneMax ขนาด 1,000 บิต	34
12	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆใน ปัญหา OneMax ขนาด 10,000 บิต	34
13	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆใน ปัญหา OneMax ขนาด 100,000 บิต	35
14	เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา OneMax	38
15	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ใน ปัญหา RandomMax ขนาด 1,000 บิต	39
16	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ใน ปัญหา RandomMax ขนาด 10,000 บิต	40
17	เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา RandomMax	43
18	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆใน ปัญหา RoyalRoad ขนาด 1,000 บิต	44
19	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆใน ปัญหา RoyalRoad ขนาด 10,000 บิต	45

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สารบัญตาราง (ต่อ)

ตารางที่		หน้า
20	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RoyalRoad ขนาด 100,000บิต	45
21	เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา RoyalRoad	48
22	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RandomRoyalRoad ขนาด 1,000 บิต	49
23	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RandomRoyalRoad ขนาด 10,000 บิต	50
24	เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา RandomRoyalRoad	53
25	ความยาวของโครโมโซม LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา OneMax	54
26	ความยาวของโครโมโซม LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา RoyalRoad	55
27	จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา OneMax	56
28	จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา RoyalRoad	57
29	จำนวนรุ่นที่พบคำตอบ	58
30	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุดในปัญหา RandomRoyalRoad	59
31	จำนวนโครโมโซมที่ shift แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมในปัญหาทดสอบ	61
32	จำนวนโครโมโซมที่ shift แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมในปัญหาทดสอบ	62

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สารบัญภาพ

ภาพที่		หน้า
1	ขั้นตอนวิธีเชิงพันธุกรรม	6
2	การทำงานของขั้นตอนวิธีเชิงพันธุกรรม	6
3	การไขว้เปลี่ยนแบบจุดเดียว	7
4	การไขว้เปลี่ยนแบบสองจุด	7
5	การไขว้เปลี่ยนแบบยูนิฟอร์ม	8
6	การกลายพันธุ์ของโครโมโซมในขั้นตอนวิธีเชิงพันธุกรรม	8
7	การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม	9
8	อัลกอริทึมการคลายของ LZW	10
9	การคลายโครโมโซมของ LZWGA	11
10	คำตอบที่ LZWGA หาได้ในปัญหา RoyalRoad	22
11	การ shift โครโมโซม LZWGA ที่ตำแหน่งต่างๆ ในปัญหา RoyalRoad	23
12	การ shift และ shift-and-check โครโมโซมที่ตำแหน่งที่ 1 ในปัญหา RoyalRoad	24
13	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา OneMax ขนาด 1,000 บิต	36
14	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา OneMax ขนาด 10,000 บิต	36
15	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา OneMax ขนาด 100,000 บิต	37
16	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomMax ขนาด 1,000 บิต	41
17	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomMax ขนาด 10,000 บิต	41
18	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RoyalRoad ขนาด 1,000 บิต	46
19	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RoyalRoad ขนาด 10,000 บิต	46

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
20	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RoyalRoad ขนาด 100,000 บิต	47
21	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomRoyalRoad ขนาด 1,000 บิต	51
22	ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomRoyalRoad ขนาด 10,000 บิต	51
23	ความยาวโครโมโซมของ LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา OneMax	54
24	ความยาวโครโมโซมของ LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา RoyalRoad	55
25	จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา OneMax	57
26	จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา RoyalRoad	58
27	ค่าความเหมาะสมของโครโมโซมของ LZWGA ที่ได้ในปัญหา OneMax	59
28	ค่าความเหมาะสมของโครโมโซมของ LZWGA ที่ได้ในปัญหา RoyalRoad	60
29	ค่าความเหมาะสมของโครโมโซมของ LZWGA ที่ได้ในปัญหา RandomRoad	60
30	ความน่าจะเป็นที่การ shift ทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้น	63
31	ความน่าจะเป็นที่การ shift-and-check ทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้น	63

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

คำอธิบายสัญลักษณ์และคำย่อ

GA	=	Genetic Algorithm
LZW	=	Lempel-Ziv-Welch Algorithm
LZWGA	=	LZW Encoding in Genetic Algorithm
ISC	=	Improved shift-and-check operator
โครโมโซมที่ถูกบีบอัด	=	โครโมโซมที่เข้ารหัสและสามารถคลายได้ด้วยอัลกอริทึมการคลายของ LZW
โครโมโซมที่ถูกคลาย	=	โครโมโซมที่ผ่านการคลายด้วยอัลกอริทึมการคลายของ LZW

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตัวดำเนินการ Shift-and-Check สำหรับ LZWGA

A Shift-and-Check Operator for LZWGA

คำนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm - GA) เป็นวิธีการค้นหาคำตอบที่เลียนแบบการวิวัฒนาการของสิ่งมีชีวิต (Mitchell, 1998) โดยใช้กลไกการคัดเลือกตามธรรมชาติซึ่งสิ่งมีชีวิตที่เหมาะสมเท่านั้นจึงจะมีโอกาสอยู่รอด การค้นหาคำตอบที่เหมาะสมที่สุดของขั้นตอนวิธีเชิงพันธุกรรมดำเนินการกับโครโมโซมที่ถูกเข้ารหัสเป็นเลขฐานสอง ระหว่างการวิวัฒนาการโครโมโซมจะถูกเปลี่ยนแปลงโดยตัวดำเนินการทางพันธุกรรม เช่น การไขว้เปลี่ยนและการกลายพันธุ์ จนกว่าจะพบคำตอบที่ต้องการ ขั้นตอนวิธีเชิงพันธุกรรมเหมาะกับการค้นหาคำตอบในกรณีที่ปริภูมิการค้นหามีขนาดใหญ่ซึ่งไม่สะดวกต่อการค้นหาทุกรูปแบบ แต่เมื่อโครโมโซมมีขนาดใหญ่มากๆ ปริภูมิการค้นหาก็จะมีขนาดใหญ่ขึ้นเป็นทวีคูณ เช่น ในปัญหา OneMax หนึ่งล้านบิต มีคำตอบที่เป็นไปได้ทั้งหมด $2^{1000000}$ แบบ ทำให้การหาคำตอบโดยใช้ขั้นตอนวิธีเชิงพันธุกรรมใช้เวลานานและอาจหาคำตอบที่ต้องการไม่พบ

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZW Encoding in Genetic Algorithm - LZWGA) ใช้การเข้ารหัสโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดทำให้โครโมโซมมีขนาดเล็กลงส่งผลให้ปริภูมิการค้นหามีขนาดเล็กลง โครโมโซมที่ถูกบีบอัดจะอยู่ในรูปอาร์เรย์ของจำนวนเต็ม (นริศ และ วรเศรษฐ, 2548) การทำงานของ LZWGA เพิ่มขั้นตอนการคลายโครโมโซมก่อนนำไปวัดค่าความเหมาะสม โดยการวิวัฒนาการทำกับข้อมูลที่ถูกบีบอัด เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำมาคลายจะได้โครโมโซมของ GA คือ 1010101 โครโมโซมที่ได้นี้จะถูกนำไปวัดค่าความเหมาะสมต่อไป LZWGA สามารถแก้ปัญหาที่มีขนาดใหญ่มากถึงหนึ่งล้านบิตได้ (Kunasol *et al*., 2005) ปัญหาดังกล่าวคือ OneMax, RoyalRoad, และ Trap ซึ่งคำตอบของปัญหาดังกล่าวมีแบบแผนสูง (คำตอบเป็น 1 หมดทุกบิต) แต่สำหรับปัญหาที่ไม่มีแบบแผน เช่น ปัญหา RandomMax และ RandomRoyalRoad (ที่มีค่าความสุ่มเป็น 1) LZWGA ไม่สามารถหาคำตอบพบ (สาวิตรี และ วรเศรษฐ, 2550) นอกจากนั้นยังได้คำตอบที่มีค่าความเหมาะสมเฉลี่ยน้อยกว่า GA

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

จากการทดลองใช้ LZWGA แก้ปัญหา RoyalRoad โดยกำหนดโครโมโซม LZWGA ขนาด 47 ตำแหน่ง ซึ่งสามารถคลายได้ 1,128 บิตพอดี (มีคำตอบในปริภูมิการค้นหาเพียงคำตอบเดียว) โครโมโซม LZWGA ที่เป็นคำตอบของปัญหาดังกล่าวเป็นตัวเลขที่เรียงกันตามลำดับ 1, 2, 3, …, 47 ซึ่ง LZWGA หาคำตอบไม่พบ คำตอบที่ LZWGA หาได้ยังเป็นตัวเลขที่ค่อนข้างเรียงกันแต่ยังไม่ถูกต้องคือยังมีบางตำแหน่งที่มีค่าซ้ำกันบ้าง แต่ส่วนใหญ่เป็นคำตอบที่ใกล้จะเรียงลำดับกันได้อย่างถูกต้องแล้ว หากมีการเลื่อนตำแหน่งของโครโมโซมน่าจะทำให้โครโมโซมที่ได้มีค่าความเหมาะสมเพิ่มขึ้น ดังนั้นจึงเกิดแนวความคิดในการสร้างตัวดำเนินการ shift คือการเลื่อนตำแหน่งของโครโมโซมเพื่อปรับปรุงประสิทธิภาพการค้นหาของ LZWGA

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

วัตถุประสงค์

1. เพื่อพัฒนาตัวดำเนินการที่ช่วยปรับปรุงประสิทธิภาพการทำงานของ LZWGA

1.1 เพื่อให้ LZWGA สามารถหาคำตอบพบได้โดยใช้ความยาวโครโมโซมของ LZWGA ที่สั้นลง และใช้จำนวนรุ่นในการพบคำตอบน้อยลง

1.2 เพื่อให้ LZWGA สามารถสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้นในกรณีที่หาคำตอบไม่พบ

2. เพื่อวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการ shift ใน LZWGA

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ตัวดำเนินการที่ช่วยปรับปรุงประสิทธิภาพการทำงานของ LZWGA

1.1 LZWGA สามารถหาคำตอบพบได้ แม้ในปัญหาที่กำหนดความยาวโครโมโซมให้สั้นพอดีกับคำตอบ และใช้จำนวนรุ่นในการพบคำตอบน้อย

1.2 LZWGA สามารถสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้นในกรณีที่หาคำตอบไม่พบ

2. ได้ทราบความน่าจะเป็นที่การใช้ตัวดำเนินการ shift จะทำให้ LZWGA มีประสิทธิภาพในการทำงานสูงขึ้น

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ขอบเขตและข้อจำกัด

1. เปรียบเทียบประสิทธิภาพการทำงานของ LZWGA ธรรมดา กับ LZWGA ที่ใช้ตัวดำเนินการ shift

2. ทดสอบกับปัญหา OneMax, RoyalRoad, RandomMax, และ RandomRoyalRoad

3. วัดประสิทธิภาพโดยดูจากจำนวนรุ่นที่พบคำตอบ, ความยาวโครโมโซมที่ใช้ และค่าความเหมาะสมของโครโมโซมที่ดีที่สุด

4. วิเคราะห์ประสิทธิภาพโดยดูจากความน่าจะเป็นที่การ shift จะทำให้ได้โครโมโซมที่มีค่าความเหมาะสมเพิ่มขึ้น

5. ส่วนของการวัดและการวิเคราะห์จะใช้การทดลอง ไม่ใช่การพิสูจน์ทางคณิตศาสตร์

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

การตรวจเอกสาร

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องประกอบด้วย ขั้นตอนวิธีเชิงพันธุกรรม (GA) และการเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA)

1. ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)

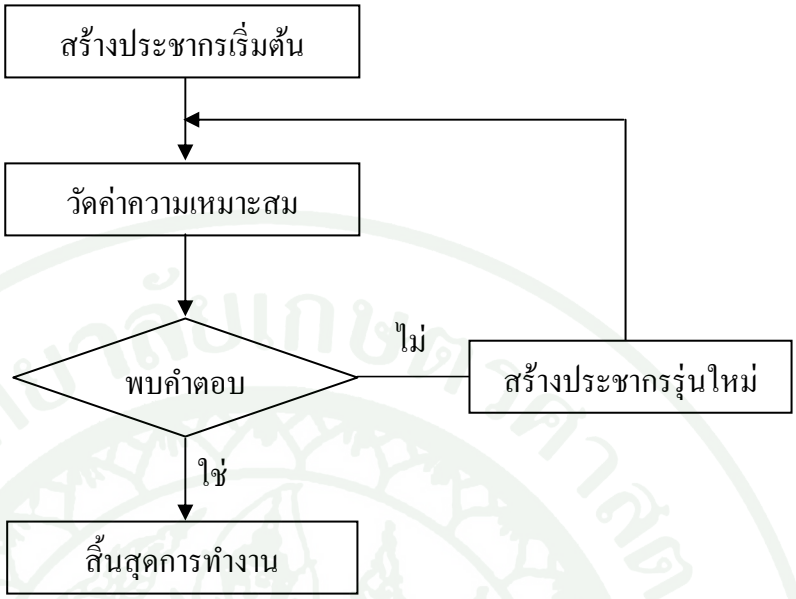
1.1 ความหมายของขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนวิธีเชิงพันธุกรรม เป็นวิธีการค้นหาคำตอบที่เลียนแบบวิวัฒนาการของสิ่งมีชีวิต (Mitchell, 1998) โดยใช้กลไกการคัดเลือกตามธรรมชาติ (natural selection) การค้นหากระทำกับกลุ่มของโครโมโซมที่เป็นเลขฐานสอง (binary string) เมื่อยังไม่พบคำตอบก็จะมีการวิวัฒนาการไปเรื่อยๆ โดยใช้ตัวดำเนินการทางพันธุกรรม คือการไขว้เปลี่ยนและการกลายพันธุ์ จนกว่าจะพบคำตอบที่ต้องการ ขั้นตอนวิธีเชิงพันธุกรรมเหมาะกับการค้นหาคำตอบในกรณีที่ปริภูมิการค้นหามีขนาดใหญ่ ไม่สามารถค้นหาทุกรูปแบบ (exhaustive search) ขั้นตอนวิธีเชิงพันธุกรรมเป็นเทคนิคในการค้นหาข้อมูลที่ได้ผลดีและได้รับความสนใจนำไปประยุกต์ใช้กับงานด้านต่างๆ อย่างกว้างขวาง

1.2 การทำงานของขั้นตอนวิธีเชิงพันธุกรรม

การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแสดงดังภาพที่ 1 โดยเริ่มต้นจะสร้างโครโมโซมโดยการสุ่ม หลังจากนั้นจะคัดเลือกโครโมโซมจากค่าความเหมาะสมและนำมาสร้างโครโมโซมรุ่นใหม่โดยใช้ตัวดำเนินการทางพันธุกรรมเพื่อให้เกิดคำตอบใหม่ๆ

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 1 ขั้นตอนวิธีเชิงพันธุกรรม

รายละเอียดการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแสดงดังภาพที่ 2 โดยจะมีการวนรอบไปเรื่อยๆ จนกว่าจะพบคำตอบหรือครบเงื่อนไขของการสิ้นสุดการทำงาน

1. สุ่มสร้างโครโมโซมเลขฐานสองจำนวน n โครโมโซม
2. วัดค่าความเหมาะสมของโครโมโซมแต่ละตัว
3. โครโมโซมที่ได้เป็นคำตอบที่ดีที่สุดแล้วหรือไม่ ถ้าใช่ จบการทำงาน
4. สร้างโครโมโซมรุ่นใหม่โดยมีกระบวนการ คือ
   4.1 การไขว้เปลี่ยน นำโครโมโซมที่ได้จากขั้นตอนที่ 3 มาทำการผสมเพื่อให้ได้โครโมโซมรุ่นใหม่
   4.2 การกลายพันธุ์ โดยสุ่มเปลี่ยนแปลงบางบิตในโครโมโซม
5. ทำซ้ำขั้นตอนที่ 2

ภาพที่ 2 การทำงานของขั้นตอนวิธีเชิงพันธุกรรม

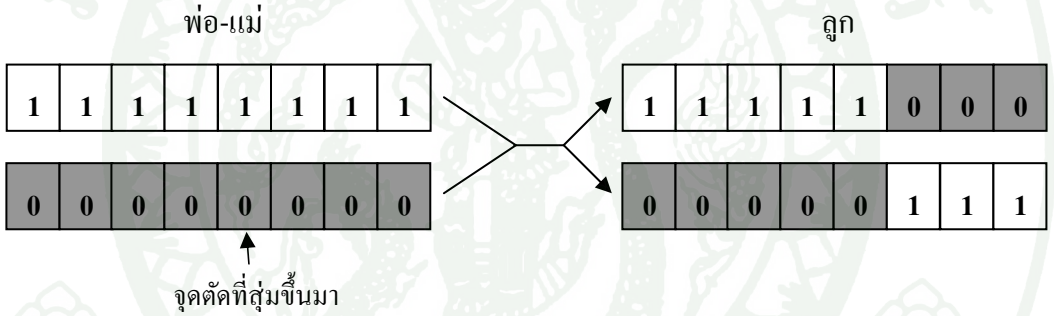
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

1.3 ตัวดำเนินการทางพันธุกรรม (Genetic Operator)

การสร้างประชากรรุ่นใหม่ของขั้นตอนวิธีเชิงพันธุกรรมต้องใช้ตัวดำเนินการทางพันธุกรรมซึ่งประกอบด้วย การไขว้เปลี่ยนและการกลายพันธุ์

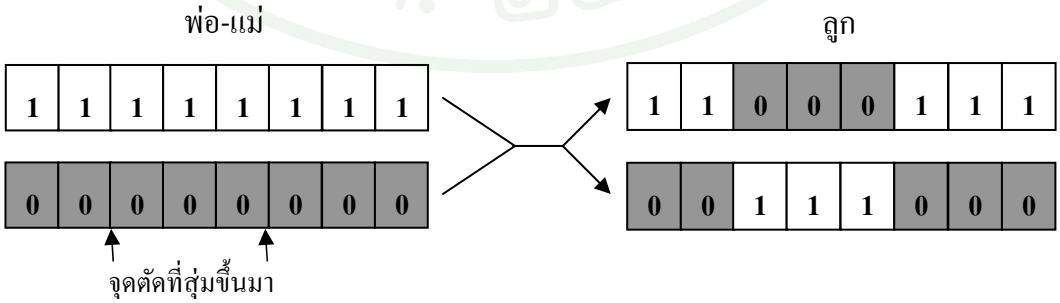
1.3.1 การไขว้เปลี่ยน (Crossover) เป็นการนำโครโมโซมที่ถูกคัดเลือกมาผสมกันเพื่อให้ได้โครโมโซมรุ่นใหม่ วิธีการไขว้เปลี่ยนโดยทั่วไปประกอบด้วย การไขว้เปลี่ยนแบบจุดเดียว, การไขว้เปลี่ยนแบบสองจุด, และการไขว้เปลี่ยนแบบยูนิฟอร์ม

(1) การไขว้เปลี่ยนแบบจุดเดียว (one-point crossover) เป็นการสุ่มจุดตัดเพียงจุดเดียวและไขว้เปลี่ยนโครโมโซมหลังจุดตัดนั้นดังภาพที่ 3



ภาพที่ 3 การไขว้เปลี่ยนแบบจุดเดียว

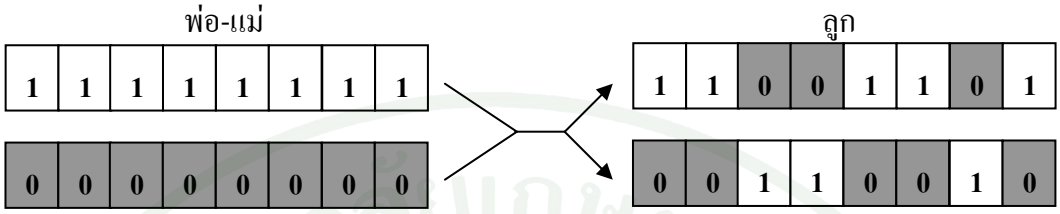
(2) การไขว้เปลี่ยนแบบสองจุด (two-point crossover) เป็นการสุ่มจุดตัดสองจุดแล้วไขว้เปลี่ยนโครโมโซมระหว่างจุดตัดทั้งสองดังภาพที่ 4



ภาพที่ 4 การไขว้เปลี่ยนแบบสองจุด

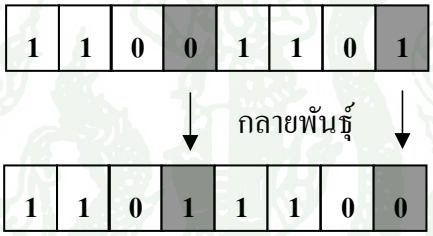
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

(3) การไขว้เปลี่ยนแบบยูนิฟอร์ม (uniform crossover) เป็นนำโครโมโซม 2 ตัว มาสุ่มแลกเปลี่ยนโครโมโซมเฉพาะตำแหน่งที่สุ่มดังภาพที่ 5



ภาพที่ 5 การไขว้เปลี่ยนแบบยูนิฟอร์ม

1.3.2 การกลายพันธุ์ (Mutation) เป็นการเปลี่ยนแปลงค่าบิตภายในโครโมโซมของตนเองอย่างสุ่ม โดยจะแปลงค่าบิตปัจจุบันเป็นตรงข้าม คือ จากบิต 0 เป็นบิต 1 และจากบิต 1 เป็นบิต 0 การกลายพันธุ์แสดงดังภาพที่ 6



ภาพที่ 6 การกลายพันธุ์ของโครโมโซมในขั้นตอนวิธีเชิงพันธุกรรม

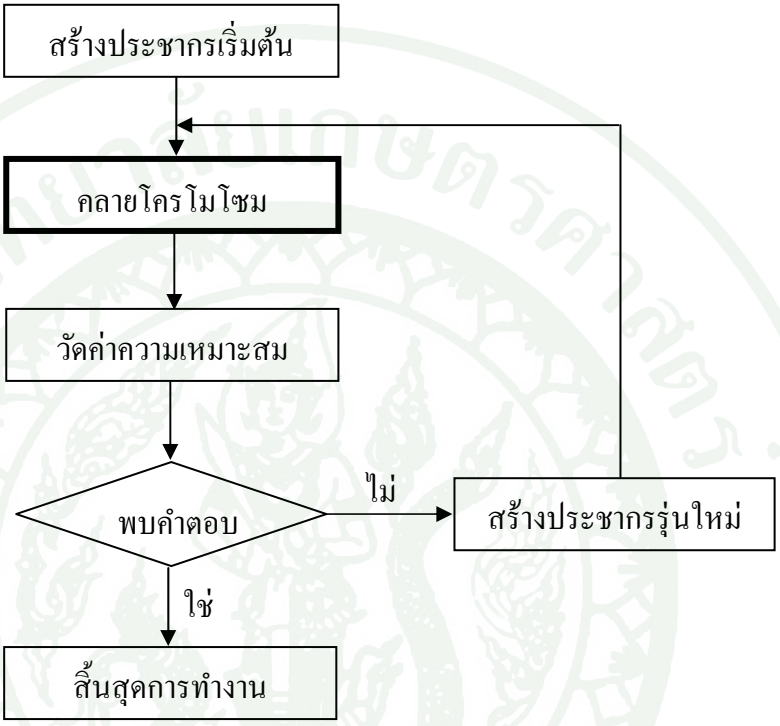
2. การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZW Encoding in Genetic Algorithm)

2.1 ความหมายของการเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม ใช้การเข้ารหัสโครโมโซมของขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัดทำให้โครโมโซมมีขนาดเล็กลงเพื่อลดขนาดของปริภูมิการค้นหา (นริศ และคณะ, 2548) การทำงานของ LZWGA เพิ่มส่วนของการคลายโครโมโซมก่อนนำไปวัดค่าความเหมาะสม โดยการวิวัฒนาการทำกับข้อมูลที่ถูกบีบอัด เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำมาคลายจะได้โครโมโซมของ GA คือ 1010101 โครโมโซมที่ได้นี้จะถูกนำไปวัดค่าความเหมาะสมต่อไป

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

โครโมโซมของ LZWGA เป็นอาร์เรย์ของเลขจำนวนเต็ม การทำงานของ LZWGA ต่างจาก GA ตรงที่เพิ่มขั้นตอนการคลายโครโมโซมเข้ามาก่อนนำไปวัดค่าความเหมาะสม ผังของการทำงานแสดงดังภาพที่ 7



ภาพที่ 7 การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม

2.2 การคลายและการวัดค่าความเหมาะสม

จากอัลกอริทึมการบีบอัดข้อมูลของ LZW (Yokoo, 1992) ขณะที่บีบอัดจะมีการสร้าง dictionary เก็บค่าสายบิตที่ยังไม่เคยเจอมาก่อน (โครโมโซมเป็นเลขฐานสอง ค่าเริ่มต้นจึงเป็น 0 และ 1) ค่าคีย์ใน dictionary จะถูกส่งออกเป็นผลการบีบอัด เมื่อเอาชุดข้อมูลที่ถูกบีบอัดนี้ไปทำการคลาย LZW ก็จะทำการสร้าง dictionary ขึ้นมาใหม่ที่เหมือนกับตัวเดิมทุกประการเพื่อเอาข้อมูลที่ค่าคีย์นั้นอ้างถึงใน dictionary มาสร้างเป็นชุดข้อมูลสายอักขระที่เหมือนตัวข้อมูลเดิมก่อนทำการบีบอัด อัลกอริทึมของการคลายจะแสดงในภาพที่ 8

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

1. เพิ่มค่า 0 และ 1 ลงใน dictionary
2. อ่านค่า *p* จาก dictionary และ เอาต์พุต *p* ออกมา
3. วนรอบเท่าจำนวนบิตในโครโมโซม อ่านค่า *c* ทีละอักขระจากโครโมโซมที่ต้องการคลาย
4. ถ้าไม่เคยมี *c* อยู่ใน dictionary
5. เพิ่ม *str* (*p*) + *fc* (*str* (*p*)) ลงใน dictionary
6. เอาต์พุตของการคลายอักขระ *c* คือ *str* (*p*) + *fc* (*str* (*p*))
7. ถ้ามี *c* อยู่ใน dictionary
8. เพิ่ม *str* (*p*) +*fc* (*str* (*c*)) ลงใน dictionary
9. เอาต์พุตของการคลายอักขระ *c* คือ *str* (*c*)
10. *p* = *c*;

ภาพที่ 8 อัลกอริทึมการคลายของ LZW

เมื่อ *fc* (*x*) คือ อักษรตัวแรกของสายบิตใน dictionary ถูกอ้างถึงโดยคีย์ *x*
str (*x*) คือ สายบิตใน dictionary ถูกอ้างถึงโดยคีย์ *x*
c คือ ค่าอักขระที่อ่านได้จากโครโมโซมที่ถูกบีบอัดอยู่
p คือ ค่าตัวแปรที่รอเก็บค่า *c* และจะเป็นตัวที่ถูกอ้างถึงในการคลายบิตถัดไป กำหนดค่าเริ่มต้นเป็นอักษรตัวแรกที่อ่านเข้ามาจากสายบิต

ผลที่ได้จากการคลายจะถูกนำไปวัดค่าความเหมาะสม โดยวัดแบบเดียวกับโครโมโซมของ GA ตัวอย่างการคลายโครโมโซมของ LZW แสดงดังภาพที่ 9 โดยกำหนดให้โครโมโซมที่ต้องการคลาย คือ {0, 2, 1, 3, 1, 1} (นริศ และ วรเศรษฐ, 2548)

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

การคลาย		dictionary		
chromosome	**output**	**index**	**full string**	
		0	**0**	ค่าเริ่มต้น
		1	**1**	
0	0	-	-	
2	00	2	00	
1	1	3	001	
3	001	4	10	
1	1	5	0011	
1	1	6	11	

โครโมโซม {0, 2, 1, 3, 1, 1} ทำการคลายด้วย LZW ได้ผลลัพธ์คือ 000100111

ภาพที่ 9 การคลายโครโมโซมของ LZWGA

2.3 เงื่อนไขในการสร้างประชากรรุ่นแรก

อัลกอริทึมการคลายของ LZW ไม่สามารถคลายโครโมโซมบางตัวที่สร้างขึ้นโดยการสุ่มได้ เนื่องจากในขั้นของการคลายต้องมีการอ้างถึงค่าคีย์ *p* ที่มีอยู่ใน dictionary โดยค่าคีย์ดังกล่าวนี้กำหนดค่าเริ่มต้นเป็น 0 และ 1 หลังจากนั้นจะถูกสร้างเพิ่มขึ้นควบคู่ไปกับกระบวนการคลายโครโมโซม หากค่าคีย์ที่ถูกอ้างถึงนี้ยังไม่มีอยู่ใน dictionary ก็จะไม่สามารถคลายโครโมโซมได้ ตัวอย่างการคลายโครโมโซมเช่น เริ่มต้นคลายตำแหน่งที่ 1 ของโครโมโซม ค่าคีย์ที่สามารถอ้างถึงได้ในขณะนั้นคือ 0 และ 1 ซึ่งเป็นค่าคีย์เริ่มต้นและอัลกอริทึมจะสร้างคีย์ที่ 2 ขึ้น ต่อมาเมื่อคลายตำแหน่งที่ 2 ค่าคีย์ที่สามารถอ้างถึงได้คือ 0, 1, 2 และอัลกอริทึมจะสร้างคีย์ที่ 3 ขึ้น นั่นคือเมื่อทำการคลายตำแหน่งที่ *i* ค่าคีย์ที่สามารถอ้างถึงได้คือ 0, 1, 2, 3, .., *i* และอัลกอริทึมจะสร้างคีย์ที่ *i*+1 ขึ้น โดยคีย์ *i*+1 ที่สร้างขึ้นในรอบนี้จะสามารถถูกอ้างถึงได้ในรอบถัดไป ตัวอย่างโครโมโซมที่อัลกอริทึมของ LZW ไม่สามารถคลายได้ เช่น {1, 3, 0, 2, 4} เพราะเมื่อถึงตำแหน่งที่ 2 ค่าคีย์ที่สามารถอ้างได้คือ 0, 1, 2 ส่วนค่า 3 ไม่เคยมีอยู่ใน dictionary ดังนั้นจึงไม่สามารถคลายโครโมโซมได้

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

จากเงื่อนไขในการคลายโครโมโซมด้วยอัลกอริทึมของ LZW ดังกล่าวจะเห็นว่าค่าของโครโมโซมใน LZWGA มีความสัมพันธ์กับตำแหน่งในโครโมโซมด้วย โดยโครโมโซมตำแหน่งที่ i จะไม่สามารถมีค่าเกินค่าของ i ได้ ดังนั้นในการสร้างประชากรรุ่นแรกโครโมโซมที่ได้จากการสุ่มจะต้องอยู่ในเงื่อนไขดังสมการที่ 1

$$a_i = r \bmod (i+1),\ i \in \{1, 2, 3, \ldots, n\} \quad (1)$$

โดย a_i คือ ค่าในตำแหน่งที่ i ของโครโมโซมที่ถูกบีบอัดอยู่

r คือ ตัวเลขจำนวนเต็มที่ได้จากการสุ่ม

n คือ ความยาวของโครโมโซมที่ถูกบีบอัด

3. ปัญหาที่ใช้ในการทดลอง

3.1 OneMax

ปัญหา OneMax จำลองปัญหาการหาค่าที่เหมาะสมที่สุด (optimization problem) ที่ตัวแปรทุกตัวเป็นอิสระต่อกัน โดยใช้การนับจำนวนเลขหนึ่งในโครโมโซม ค่าความเหมาะสมที่ได้ก็คือ จำนวนเลขหนึ่งทั้งหมดที่พบในโครโมโซม ดังนั้นโครโมโซมที่ดีที่สุดคือโครโมโซมที่ทุกบิตมีค่าเป็นหนึ่ง

ตัวอย่างการวัดค่าความเหมาะสมของโครโมโซมในปัญหา OneMax คือ

โครโมโซม 1100100101010110

ค่าความเหมาะสมของโครโมโซมคือ 8

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3.2 RandomMax

ปัญหา RandomMax คล้ายกับ OneMax นั่นคือเป็นปัญหาการนับจำนวนบิตของโครโมโซมที่ตรงกับคำตอบ แต่ต่างกันตรงที่คำตอบที่ต้องการเป็นคำตอบที่เกิดจากการสุ่ม โดยค่าความเหมาะสมที่ได้คือ จำนวนบิต ณ ตำแหน่งเดียวกับคำตอบที่มีค่าเท่ากัน (สาวิตรี และ วรเศรษฐ, 2550)

ตัวอย่างการวัดค่าความเหมาะสมของโครโมโซมในปัญหา RandomMax คือ

คำตอบที่สุ่มมา	10100101
โครโมโซม	10000100

ค่าความเหมาะสมที่ได้คือจำนวนบิตที่มีค่าตรงกับคำตอบซึ่งมีค่าเท่ากับ 6 เนื่องจากในบิตที่ 3 และ 8 (จากซ้าย) มีค่าไม่ตรงกับคำตอบที่สุ่มมา สาเหตุที่นำปัญหา RandomMax มาใช้ในการทดลอง เพราะเป็นปัญหาที่ค่อนข้างยาก ใช้การสุ่มคำตอบเพื่อให้ได้คำตอบที่ไม่มีแบบแผน ซึ่งจะช่วยวัดประสิทธิภาพการทำงานของ LZWGA ได้

3.3 RoyalRoad

ปัญหา RoyalRoad จำลองปัญหา optimization ที่บิตในโครโมโซมมีความสัมพันธ์กัน โครโมโซมถูกแบ่งเป็นบล็อก คำตอบที่ต้องการคือทุกบิตในบล็อกมีค่าเท่ากับ 1 หากบิตใดบิตหนึ่งในบล็อกมีค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันทีแต่หากทุกบิตในบล็อกเป็น 1 หมด ค่าความเหมาะสมจะเท่ากับจำนวนบิตในบล็อกนั้น

ตัวอย่างค่าความเหมาะสมของโครโมโซมในปัญหา RoyalRoad บล็อก 4 บิต คือ

โครโมโซม	1111 1001 1111 0110

บล็อกที่ 1 และบล็อกที่ 3 มีค่าเป็น 1 ทั้งบล็อก ส่วนบล็อกที่ 2 และ 4 มีบางบิตมีค่าเป็น 0 ดังนั้นค่าความเหมาะสมที่ได้จากโครโมโซมนี้คือ 8

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3.4 RandomRoyalRoad

ปัญหา RandomRoyalRoad คล้ายกับ RoyalRoad คือโครโมโซมถูกแบ่งออกเป็นบล็อก แต่ต่างกันตรงที่คำตอบของแต่ละบล็อกเกิดจากการสุ่ม (สาวิตรี และ วรเศรษฐ, 2550) ค่าของบิตในบล็อกที่เป็นคำตอบจึงมีได้ทั้ง 0 และ 1 ถ้าทุกบิตในบล็อกของโครโมโซมที่ต้องการวัดค่าความเหมาะสมมีค่าเหมือนกับคำตอบ ค่าความเหมาะสมที่ได้จะเท่ากับจำนวนบิตในบล็อกนั้น แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่เหมือนกับคำตอบ บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันที

ตัวอย่างการวัดค่าความเหมาะสมของโครโมโซมในปัญหา RandomRoyalRoad คือ

คำตอบ	1101 1101 1101 1101
โครโมโซม	1101 1110 1101 1101

เนื่องจากในบล็อกที่ 1, 3, และ 4 มีค่าตรงกับคำตอบ แต่ในบล็อกที่ 2 มีบางบิตมีค่าไม่ตรงกับคำตอบ ดังนั้นค่าความเหมาะสมที่ได้จึงมีค่าเท่ากับ 12 สาเหตุที่นำปัญหา RandomRoyalRoad มาใช้ในการทดลอง เพราะเป็นปัญหาที่ค่อนข้างยากเช่นเดียวกับ RandomMax ใช้การสุ่มบล็อกของคำตอบเพื่อให้ได้คำตอบที่ไม่มีแบบแผน ซึ่งจะช่วยวัดประสิทธิภาพการทำงานของ LZWGA ได้เช่นกัน

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

งานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้มุ่งศึกษาการพัฒนาตัวดำเนินการสำหรับ LZWGA ดังนั้นส่วนของการศึกษางานวิจัยที่เกี่ยวข้องจึงมุ่งศึกษางานวิจัยที่เกี่ยวกับการบีบอัดโครโมโซมและงานวิจัยที่เกี่ยวกับตัวดำเนินการของ GA และ LZWGA ซึ่งมีรายละเอียดดังนี้

1. งานวิจัยที่เกี่ยวกับการบีบอัดโครโมโซม

งานวิจัยของ วรเศรษฐ และคณะ (2548) เสนอขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด (Compressed Genetic Algorithm - CompressedGA) ซึ่งใช้การเข้ารหัสโครโมโซมแบบบีบอัดที่ระบุจำนวนครั้งการซ้ำกันของช่วงข้อมูล ผลการทดลองพบว่า CompressedGA ใช้จำนวนรุ่นการค้นหาน้อยกว่า GA ในปัญหา OneMax ขนาด 128 บิต ถึง 805 เท่า และปัญหาแขนหุ่นยนต์ถึง 4.5 เท่า

งานวิจัยของ นริศ และคณะ (2548) นำเสนอการเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZW Encoding in Genetic Algorithm - LZWGA) โดยการเข้ารหัสโครโมโซมแบบบีบอัดเพื่อลดขนาดของปริภูมิการค้นหา โครโมโซมที่ถูกบีบอัดก่อนนำมาผ่านกระบวนการทางพันธุกรรม จะต้องนำมาคลายโดยใช้อัลกอริทึมของ Lempel-Ziv-Welch (LZW) แล้วจึงนำมาวัดค่าความเหมาะสม ประสิทธิภาพของ LZWGA ถูกนำมาเปรียบเทียบกับ GA และ CompressedGA พบว่า LZWGA ใช้จำนวนรุ่นในการแก้ปัญหา OneMax ขนาด 256 บิต น้อยกว่า GA ถึง 205 เท่า และ LZWGA ใช้จำนวนรุ่นในการแก้ปัญหา OneMax ขนาด 2048 บิต น้อยกว่า CompressedGA ถึง 14 เท่า

งานวิจัยของ Suwannik and Chongstitvatana (2007) นำเสนอ Estimation Distribution Algorithm with Arithmetic Coding (EDAAC) โดยใช้การเข้ารหัสโครโมโซมด้วยหลักการของ Arithmetic Coding ตัวดำเนินการทางพันธุกรรมที่สามารถใช้ได้กับการเข้ารหัสนี้คือการกลายพันธุ์เพียงอย่างเดียว ผลการทดลองกับปัญหา OneMax, Royal Road, Trap ขนาดหนึ่งล้านบิตพบว่า EDAAC ใช้จำนวนรุ่นเฉลี่ย 16, 14 และ 18 รุ่นตามลำดับ โดยการแก้ปัญหา OneMax ขนาดหนึ่งล้านบิต ใช้เวลา 4 วินาที และแก้ปัญหา OneMax ขนาดหนึ่งพันล้านบิตโดยใช้เวลาเพียง 92 นาที

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

2. งานวิจัยที่เกี่ยวกับการดำเนินการของ GA และ LZWGA (GA and LZWGA Operation)

2.1 งานวิจัยที่เกี่ยวกับการคัดเลือก

งานวิจัยของ Zhong *et al*. (2005) ศึกษาการเปรียบเทียบประสิทธิภาพวิธีการคัดเลือก (selection) ใน GA โดยเปรียบเทียบระหว่าง การคัดเลือกแบบวงล้อรูเล็ตต์ (roulette wheel selection) และการคัดเลือกแบบประลอง (tournament selection) โดยทดลองกับฟังก์ชันทดสอบ 7 ฟังก์ชัน เปรียบเทียบการคัดเลือกแบบต่างๆ โดยใช้การไขว้เปลี่ยนและการกลายพันธุ์แบบเดียวกัน ผลการทดลองพบว่าการคัดเลือกแบบประลองดีกว่าการคัดเลือกแบบวงล้อรูเล็ตต์ ช่วยให้ค้นพบคำตอบได้เร็วกว่าและใช้เวลาในการทำงานน้อยกว่าในทุกปัญหาทดสอบ

งานวิจัยของ Jadaan *et al*. (2005) เสนอการปรับปรุงวิธีการคัดเลือกแบบวงล้อรูเล็ตต์ โดยตั้งชื่อว่า Rank Based Roulette Wheel Selection (RRWS) ซึ่งแต่ละโครโมโซมถูกกำหนดค่าความเหมาะสมให้เท่ากับอันดับ (rank) ของมัน การทดลองทำการประเมินประสิทธิภาพของ RRWS เทียบกับการคัดเลือกแบบวงล้อรูเล็ตต์ การทดสอบทำกับฟังก์ชันทดสอบ 8 ฟังก์ชัน ผลการทดลองพบว่า RRWS ช่วยให้พบคำตอบได้เร็วกว่าวงล้อรูเล็ตต์เสมอ และยังช่วยเพิ่มความหลากหลายให้กับโครโมโซมด้วย

2.2 งานวิจัยที่เกี่ยวกับการไขว้เปลี่ยนและการกลายพันธุ์

งานวิจัยของ Rasheed (1999) พัฒนาตัวดำเนินการตัวใหม่สำหรับ GA ชื่อว่า Guided Crossover (GC) ตัวดำเนินการดังกล่าวมีหลักการทำงานคือ จะเลือกโครโมโซมตัวแรกที่มีค่าความเหมาะสมมากที่สุดมาเป็นพ่อแม่ ในขณะที่โครโมโซมอีกตัวจะเลือกโดยใช้สมการวัดระยะห่างจากโครโมโซมตัวที่หนึ่งโดยเลือกให้ห่างจากโครโมโซมตัวแรก เพื่อให้ลูกที่เกิดใหม่มีความหลากหลายของโครโมโซมสูง จากนั้นจะไขว้เปลี่ยนแบบมีเงื่อนไขตามหลักการของ GC ทำการทดลองใช้ GC กับปัญหาทดลอง 2 ปัญหา คือ Supersonic Transport Aircraft Design และ Three-stage Membrane Separation Process ผลการทดลองพบว่า GC สามารถช่วยในการปรับปรุง GA ได้มากโดยจะช่วยให้ใช้การวนรอบหาคำตอบน้อยลง

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

งานวิจัยของ Feng and Yang (2002) เสนอการปรับปรุงการกลายพันธุ์ด้วยการใช้ตัวดำเนินการ Exclusive OR และ Exclusive NOR โดยคัดเลือกโครโมโซมที่จะเป็นพ่อแม่มาก่อน แล้วสุ่มเลือกตำแหน่งที่ต้องการกลายพันธุ์ เมื่อได้ตำแหน่งที่ต้องการให้เปรียบเทียบค่าของบิตในโครโมโซมทั้งสองตัวถ้าค่าเหมือนกันให้เปลี่ยนเป็น 0 เมื่อใช้ Exclusive OR และ เป็น 1 เมื่อใช้ Exclusive NOR ถ้าต่างกันก็เป็นตรงกันข้าม ผลการทดลองพบว่าการกลายพันธุ์แบบนี้ช่วยพัฒนาประสิทธิภาพของ GA ได้มากขึ้น

งานวิจัยของ Wang and Okazaki (2007) เสนอการปรับปรุงประสิทธิภาพของ GA โดยใช้การไขว้เปลี่ยนและการกลายพันธุ์แบบมีเงื่อนไขแทนที่จะกำหนดเป็นค่าความน่าจะเป็นเหมือน GA ธรรมดา ซึ่งอัตราการกลายพันธุ์แบบมีเงื่อนไขนี้ใช้การหาค่าความแตกต่างระหว่างพ่อแม่เทียบกับขนาดของโครโมโซม (difference degree) การทดลองใช้ตัวดำเนินการแบบมีเงื่อนไขนี้เปรียบเทียบกับการใช้ตัวดำเนินการที่กำหนดค่าความน่าจะเป็นแบบเดิมโดยทดสอบกับปัญหาทดสอบทั้งหมด 5 ฟังก์ชันที่มีระดับความยากง่ายปะปนกัน นอกจากนี้ยังนำไปประยุกต์ใช้กับปัญหา Set-Covering Problem (SCP) ผลการทดลองพบว่าการใช้ตัวดำเนินการแบบมีเงื่อนไขช่วยเพิ่มประสิทธิภาพการทำงานของ GA ได้และเหมาะกับการนำไปประยุกต์ใช้ในการแก้ปัญหาต่างๆ มากกว่าการใช้ตัวดำเนินการโดยกำหนดค่าความน่าจะเป็น

งานวิจัยของ Numnark and Suwannik (2008) เสนอการปรับปรุงการกลายพันธุ์ใน LZWGA โดยมีแนวความคิดมาจากการกลายพันธุ์ใน LZWGA แต่ละตำแหน่งส่งผลไม่เท่ากัน อัตราการกลายพันธุ์ของโครโมโซมใน LZWGA จึงควรถูกแบ่งเป็นช่วงตามตำแหน่งของโครโมโซม โดยให้ช่วงแรกของโครโมโซมมีโอกาสเกิดการกลายพันธุ์น้อยและช่วงหลังค่อยๆ เพิ่มโอกาสในการเกิดการกลายพันธุ์มากขึ้น ซึ่งการปรับปรุงการกลายพันธุ์ใน LZWGA โดยพิจารณาจากตำแหน่งของโครโมโซมดังกล่าวสามารถแก้ปัญหา One Max ขนาดหนึ่งล้านบิตโดยใช้จำนวนรุ่นน้อยกว่าการกลายพันธุ์ปกติ 2.2 เท่า และแก้ปัญหา Trap โดยใช้จำนวนรุ่นน้อยกว่าการกลายพันธุ์ปกติ 1.6 เท่า

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สรุปงานวิจัยที่เกี่ยวข้องเรียงตามประเภทงานวิจัย (ไทย-ต่างประเทศ) และเรียงตามปีที่พิมพ์

ที่	ปี	ชื่อบทความ	สิ่งที่นำเสนอ	ผลการทดลอง
1	2548	ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด	เข้ารหัสโครโมโซมแบบบีบอัดในรูปแบบเลขฐานสอง (Compressed GA) โดยอาศัยการซ้ำกันของช่วงข้อมูลเพื่อลดขนาดของปริภูมิการค้นหา	CompressedGA ใช้จำนวนรุ่นการค้นหาน้อยกว่า GA ในปัญหา OneMax ขนาด 128 บิต 805 เท่า และปัญหาแขนหุ่นยนต์ 4.5 เท่า
2	2548	การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม	เข้ารหัสโครโมโซมแบบบีบอัดในรูปแบบอาร์เรย์ของจำนวนเต็มที่คลายได้ด้วยอัลกอริทึมของ LZW (LZWGA)	LZWGA ใช้จำนวนรุ่นในการแก้ปัญหา OneMax ขนาด 256 บิต น้อยกว่า GA 205 เท่า และ LZWGA ใช้จำนวนรุ่นในการแก้ปัญหา OneMax ขนาด 2048 น้อยกว่า CompressedGA 14 เท่า
3	1999	Guided Crossover: A New Operator for Genetic Algorithm Based Optimization	ตัวดำเนินการ Guided Crossover (GC) เลือกโครโมโซมที่จะนำมาเป็นพ่อแม่ โดยตัวแรกเลือกตัวที่ดีที่สุดและอีกตัวเลือกโดยวัดระยะห่างจากโครโมโซมตัวที่หนึ่งนำมาทำการไขว้เปลี่ยนแบบมีเงื่อนไข	GC สามารถช่วยในการปรับปรุง GA ได้โดยจะช่วยให้ใช้การวนรอบหาคำตอบน้อยลง

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ที่	ปี	ชื่อบทความ	สิ่งที่นำเสนอ	ผลการทดลอง
4	2002	Improved Genetic Operator for Genetic Algorithm	ปรับปรุงการกลายพันธุ์ด้วยการใช้ตัวดำเนินการ Exclusive OR และ Exclusive NOR เปรียบเทียบค่าของบิตที่ต้องการกลายพันธุ์ในโครโมโซมทั้งสองตัวถ้าค่าเหมือนกันให้เปลี่ยนเป็น 0 เมื่อใช้ Exclusive OR และ เป็น 1 เมื่อใช้ Exclusive NOR ถ้าต่างกันก็เป็นตรงกันข้าม	ผลการทดลองพบว่าการกลายพันธุ์แบบนี้ช่วยพัฒนาประสิทธิภาพของ GA ได้มากขึ้น
5	2005	Improved Selection Operator for GA	ปรับปรุงวิธีการคัดเลือกตั้งชื่อว่า Rank Based Roulette Wheel Selection (RRWS) แต่ละโครโมโซมถูกกำหนดค่าความเหมาะสมให้เท่ากับอันดับ (rank) ของมัน	RRWS ช่วยให้พบคำตอบได้เร็วกว่าวงล้อรูเล็ตต์เสมอ และยังช่วยเพิ่มความหลากหลายให้กับโครโมโซมด้วย
6	2005	Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms.	เปรียบเทียบระหว่าง การคัดเลือกแบบวงล้อรูเล็ตต์ (roulette wheel selection) และการคัดเลือกแบบประลอง (tournament selection)	การคัดเลือกแบบประลองดีกว่าการคัดเลือกแบบวงล้อรูเล็ตต์ ช่วยให้ค้นพบคำตอบได้เร็วกว่าและใช้เวลาในการทำงานน้อยกว่าในทุกปัญหาทดสอบ

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ที่	ปี	ชื่อบทความ	สิ่งที่นำเสนอ	ผลการทดลอง
7	2007	An Improved Genetic Algorithm with Conditional Genetic Operators and its Application to Set-Covering Problem	ปรับปรุงประสิทธิภาพของ GA โดยใช้การไขว้เปลี่ยนและการกลายพันธุ์แบบมีเงื่อนไข แทนการกำหนดเป็นค่าความน่าจะเป็นเหมือน GA ธรรมดา	การใช้ตัวดำเนินการแบบมีเงื่อนไขช่วยเพิ่มประสิทธิภาพการทำงานของ GA ได้และเหมาะกับการนำไปประยุกต์ใช้ในการแก้ปัญหาจริงมากกว่าการใช้ตัวดำเนินการโดยกำหนดค่าความน่าจะเป็น
8	2007	Improving the Performance of LZWGA by Using a New Mutation Method	ปรับปรุงการกลายพันธุ์ใน LZWGA โดยแบ่งอัตราการเกิดการกลายพันธุ์ออกเป็นช่วงตามตำแหน่งของโครโมโซม	การปรับปรุงการกลายพันธุ์สามารถแก้ปัญหา One Max ขนาดหนึ่งล้านบิตโดยใช้จำนวนรุ่นน้อยกว่าการกลายพันธุ์ปกติ 2.2 เท่า และแก้ปัญหา Trap โดยใช้จำนวนรุ่นน้อยกว่าการกลายพันธุ์ปกติ 1.6 เท่า
9	2007	Solving Large Scale Problems using Estimation Distribution Algorithm with Arithmetic Coding	Estimation Distribution Algorithm with Arithmetic Coding (EDAAC) ที่เข้ารหัสโครโมโซมด้วยหลักการของ Arithmetic Coding ใช้ตัวดำเนินการอย่างเดียวคือการกลายพันธุ์	ผลการทดลองกับปัญหา OneMax, Royal Road, Trap ขนาดหนึ่งล้านบิตพบว่า EDAAC ใช้จำนวนรุ่นเฉลี่ย 16, 14 และ 18 รุ่นตามลำดับ

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

อุปกรณ์และวิธีการ

อุปกรณ์

1. Hardware

Intel(R) Core(TM)2 Duo CPU T5850 @2.16GHz 2.17 GHz, 2 GB of RAM, 250 GB

2. Software

2.1 Java Development Toolkit: JDK1.6

2.2 NetBeans IDE 6.1

2.3 Operating System: Microsoft Windows Vista

วิธีการ

จุดมุ่งหมายของงานวิจัยนี้คือ การเพิ่มประสิทธิภาพการทำงานให้กับ LZWGA โดยการสร้างตัวดำเนินการใหม่ซึ่งความเป็นมาและวิธีการทำงานและวัดประสิทธิภาพของตัวดำเนินการ shift แสดงดังต่อไปนี้

1. ความเป็นมาของตัวดำเนินการ Shift

จากการทดลองใช้ LZWGA แก้ปัญหา RoyalRoad โดยใช้โครโมโซม LZWGA ขนาด 47 ตำแหน่ง ซึ่งสามารถคลายได้จำนวนบิตมากสุด 1,128 บิตพอดี โครโมโซม LZWGA ที่เป็นคำตอบของปัญหาดังกล่าวเป็นตัวเลขที่เรียงกันตามลำดับ 1, 2, 3,… แต่คำตอบที่ LZWGA หาได้ยังเป็นตัวเลขที่เรียงกันไม่ถูกต้องคือบางตำแหน่งมีค่าซ้ำกัน ภาพที่ 10 แสดงคำตอบที่ LZWGA หาได้จากการทดลอง 3 ครั้ง ซึ่งไม่ใช่ตัวเลขเรียงกันในตำแหน่งที่ถูกต้อง ทำให้ไม่ใช่คำตอบที่ดีที่สุด

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

คำตอบ	1	2	3	4	5	6	7	8	9	10	11

ครั้งที่ 1	1	2	3	3	3	5	6	6	7	9	9
ครั้งที่ 2	0	1	2	3	4	5	6	7	8	9	10
ครั้งที่ 3	1	2	3	4	5	6	7	8	9	9	10

ภาพที่ 10 คำตอบที่ LZWGA หาได้ ในปัญหา RoyalRoad

คำตอบที่ LZWGA หาได้ เป็นคำตอบที่ใกล้จะเรียงกันอย่างถูกต้องแล้ว หากมีการเลื่อนตำแหน่งของโครโมโซมน่าจะทำให้โครโมโซมที่ได้มีค่าความเหมาะสมเพิ่มขึ้น ดังนั้นจึงเกิดแนวความคิดในการสร้างตัวดำเนินการ shift คือการเลื่อนตำแหน่งของโครโมโซมเพื่อปรับปรุงประสิทธิภาพการค้นหาของ LZWGA

2. ตัวดำเนินการ Shift และ Shift-and-Check

2.1 ตัวดำเนินการ Shift

การ shift คือการเลื่อนส่วนของโครโมโซมหลังตำแหน่งที่สุ่มได้ทั้งหมดมาทางด้านซ้ายหนึ่งตำแหน่ง (อณิมา และ วรเศรษฐ, 2552) โดยมีขั้นตอนดังนี้

(1) สุ่มเลือกตำแหน่งที่ i เพื่อ shift โดย $i < n$ และ n คือ ความยาวโครโมโซมของ LZWGA

(2) เริ่ม shift โดยเลื่อนโครโมโซมมาทางซ้าย 1 ตำแหน่ง ดังนี้

- แทนที่ตำแหน่งที่ i ด้วยตำแหน่งที่ i+1
- แทนที่ตำแหน่งที่ i+1 ด้วยตำแหน่งที่ i+2
- …
- แทนที่ตำแหน่งที่ n–1 ด้วยตำแหน่งที่ n

โดยตำแหน่งที่ n ซึ่งเป็นตำแหน่งสุดท้ายของโครโมโซมเราจะคงค่าเดิมไว้ ไม่ทำการเปลี่ยนแปลง

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

(3) เนื่องจากโครโมโซมของ LZWGA ไม่สามารถมีค่าเกินตำแหน่งของมันได้ เช่น ตำแหน่งที่ 2 ไม่สามารถมีค่าเกิน 2 ได้ (ค่าที่เป็นไปได้คือ 0, 1, 2) การ shift อาจจะทำให้โครโมโซมอยู่ในตำแหน่งที่ไม่สามารถคลายได้ ดังนั้นโครโมโซมตำแหน่งที่เกิดการ shift จะต้องถูกนำมาผ่านเงื่อนไขดังสมการที่ 2

$$a_i = a_{i+1} \bmod (i+1) \tag{2}$$

โดย a_i คือ ตำแหน่งที่ i ของโครโมโซมที่ถูกบีบอัดอยู่

จากภาพที่ 11 แสดงตัวอย่างการ shift โครโมโซมที่ตำแหน่ง 1, 4, และ 7 ตามลำดับ โดยกำหนดให้ตำแหน่งแรกในโครโมโซมคือตำแหน่งที่ 1 จะเห็นว่าเมื่อเปลี่ยนตำแหน่งของโครโมโซมส่งผลให้ค่าความเหมาะสมที่ได้เปลี่ยนแปลงไปด้วย

ภาพที่ 11 การ shift โครโมโซม LZWGA ที่ตำแหน่งต่างๆ ในปัญหา RoyalRoad

2.2 ตัวดำเนินการ Shift -and-Check

จากภาพที่ 11 จะเห็นว่าบางครั้งการ shift ไม่ทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้น เนื่องจากก่อน shift ตำแหน่งเดิมของโครโมโซมนั้นเหมาะสมอยู่แล้ว เช่น เมื่อ shift ตำแหน่ง

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ที่ 1 ค่าความเหมาะสมที่ได้จะเป็น 0 ดังนั้น จึงเกิดแนวความคิดที่จะปรับกระบวนการ shift โดยเพิ่มเงื่อนไขที่ต้องพิจารณาคือ ถ้า $a_{i+1} \bmod (i+1) = 0$ และ $a_{i+1} > 0$ แล้ว เราจะเปลี่ยนให้ค่าของ a_i เป็นดังสมการที่ 3

$$a_i = a_{i+1} - 1 \tag{3}$$

สาเหตุที่นำค่าของ a_{i+1} ลบด้วย 1 เพื่อหลีกเลี่ยงไม่ให้ค่าของโครโมโซมตำแหน่งที่ i มีค่าเป็น 0 ซึ่งจะทำให้ค่าความเหมาะสมของโครโมโซมที่ได้ลดลง และสาเหตุที่ให้ลบด้วย 1 แทนที่จะลบด้วยเลขจำนวนเต็มอื่นก็คือจะทำให้ค่าความเหมาะสมจากการเลื่อนโครโมโซมมีโอกาสที่จะลดลงน้อยที่สุด

จากภาพที่ 12 เปรียบเทียบการ shift และ shift-and-check ที่ตำแหน่งที่ 1 ของโครโมโซม พบว่าการ shift-and-check ทำให้ได้ค่าความเหมาะสมเพิ่มขึ้น (อณิมา และ วรเศรษฐ, 2552)

ภาพที่ 12 การ shift และ shift-and-check โครโมโซมที่ตำแหน่งที่ 1 ในปัญหา RoyalRoad

2.3 การปรับปรุงตัวดำเนินการ Shift-and-Check ในการแก้ปัญหาขนาดใหญ่ (ISC)

จากแนวความคิดในการพัฒนาการกลายพันธ์แบบแบ่งช่วง (สมรักษ์ และ วรเศรษฐ, 2550) ในปัญหาขนาดใหญ่ ซึ่งกำหนดให้อัตราการกลายพันธุ์ของโครโมโซม LZWGA มีค่าแตกต่างกันไปตามแต่ตำแหน่งของโครโมโซม ทำให้ผู้วิจัยเกิดแนวความคิดที่จะสังเกตการใช้ตัว

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ดำเนินการ shift-and-check ที่ตำแหน่งต่างๆ ของโครโมโซมว่าจะส่งผลให้ค่าความเหมาะสมแตกต่างกันอย่างไร

ผู้วิจัยทำการสังเกตการทำงานของตัวดำเนินการดังกล่าวโดยสุ่มสร้างโครโมโซม LZWGA ที่มีความยาวตั้งแต่ 4 - 15 ตำแหน่ง จากนั้นทำการ shift-and-check ทุกตำแหน่งของโครโมโซม วัดค่าความเหมาะสมที่ได้หลังการ shift-and-check โดยใช้ปัญหา OneMax เราทำซ้ำการทดลองนี้ 1,000 ครั้ง ค่าความเหมาะสมของโครโมโซมของ LZWGA หลังการ shift-and-check ที่ตำแหน่งต่างๆ แสดงในตารางที่ 1

ตารางที่ 1 ค่าความเหมาะสมที่ได้หลังการ shift-and-check ที่ตำแหน่งต่างๆของโครโมโซม (*n* คือความยาวโครโมโซมของ LZWGA)

n	ตำแหน่งที่ shift-and-check														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	5.2	3.7	3.1	2.7											
5	7.3	5.2	4.5	4.0	3.7										
6	9.8	6.9	6.0	5.4	5.0	4.8									
7	12.2	9.1	8.1	7.1	6.6	6.2	6.0								
8	14.9	11.1	9.4	8.5	8.0	7.5	7.3	7.1							
9	17.1	13.0	11.3	10.3	9.6	9.2	8.8	8.5	8.3						
10	19.4	14.7	13.1	12.0	11.1	10.6	10.3	10.0	9.7	9.6					
11	22.2	16.6	15.0	13.4	12.5	12.0	11.6	11.4	11.0	10.8	10.7				
12	24.8	19.2	16.8	15.7	14.7	14.0	13.6	13.2	12.8	12.6	12.5	12.3			
13	28.8	21.9	19.2	17.3	16.4	15.6	15.2	14.7	14.4	14.1	13.8	13.6	13.5		
14	30.9	23.4	20.8	19.2	18.4	17.4	17.0	16.4	16.1	15.8	15.6	15.4	15.2	15.0	
15	34.6	26.8	23.9	21.7	20.7	19.9	19.3	18.5	18.3	18.0	17.7	17.5	17.3	17.1	17.0

จากตารางจะเห็นว่าค่าความเหมาะสมมีการเปลี่ยนแปลงตามตำแหน่งที่เกิดการ shift-and-check โดยการ shift-and-check ที่ตำแหน่งแรกส่งผลให้โครโมโซมมีค่าความเหมาะสมมากกว่าการ shift-and-check ที่ตำแหน่งหลังๆ อัตราการ shift-and-check จึงไม่ควรเท่ากันในทุกตำแหน่งของโครโมโซม แต่ควรแตกต่างกันตามแต่ตำแหน่งของมัน ดังนั้นจึงเกิดแนวความคิดในการปรับปรุงตัวดำเนินการ shift-and-check

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

การปรับปรุงตัวดำเนินการ shift-and-check หรือ ISC คือการกำหนดให้โอกาสในการเกิดการ shift-and-check ของแต่ละตำแหน่งมีมากน้อยต่างกัน ตำแหน่งแรกๆมีโอกาสในการเกิดการ shift-and-check มากกว่าตำแหน่งหลังๆ (Rotsianglum and Suwannik, 2010) ค่าของ P_i เป็นไปตามสมการที่ 4

$$P_i = \frac{n+1-i}{\sum_{j=1}^{n} j} \qquad \text{โดยที่} \quad i = 1, 2, \ldots, n \tag{4}$$

เมื่อ i คือตำแหน่งปัจจุบันของโครโมโซม

n คือ ความยาวของโครโมโซม

ตัวอย่างค่าของ P_i ในโครโมโซม LZWGA 4 ถึง 7 ตำแหน่ง แสดงในตารางที่ 2

ตารางที่ 2 ค่า P_i ของโครโมโซม LZWGA 4 ถึง 7 ตำแหน่ง

ความยาว	ตำแหน่ง						
โครโมโซม	1	2	3	4	5	6	7
4	0.40	0.30	0.20	0.10			
5	0.33	0.27	0.20	0.13	0.07		
6	0.29	0.24	0.19	0.14	0.10	0.05	
7	0.25	0.21	0.18	0.14	0.11	0.07	0.04

3. แนวทางการทดลอง

การทดลองแบ่งออกเป็น 3 ส่วน คือ การทดลองเพื่อวัดประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check, การทดลองเพื่อวัดประสิทธิภาพของการปรับปรุงตัวดำเนินการ shift-and-check และ การทดลองเพื่อวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการ shift และ shift-and-check ใน LZWGA

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3.1 การทดลองเพื่อวัดประสิทธิภาพของตัวดำเนินการ Shift และ Shift-and-Check

การทดลองเพื่อวัดประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check แบ่งการเปรียบเทียบประสิทธิภาพออกเป็น 3 กรณี คือ เปรียบเทียบจากค่าความเหมาะสม, ความยาวโครโมโซมที่ใช้ในการพบคำตอบ และ จำนวนรุ่นที่พบคำตอบ

3.1.1 การทดลองเพื่อเปรียบเทียบค่าความเหมาะสมของโครโมโซมที่ได้

การทดลองเพื่อเปรียบเทียบค่าความเหมาะสมทำกับปัญหาทดสอบ 4 ปัญหา คือ OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad ซึ่งเป็นปัญหาที่มีทั้งความง่ายและยากปะปนกัน การทดลองจะกำหนดอัตราการ shift และ shift-and-check ตั้งแต่ 0.1 ถึง 1.0 โดยการ shift จะทำหลังจากการไขว้เปลี่ยนเสร็จสิ้นแล้ว วิธีการคัดเลือกที่ใช้คือการคัดเลือกแบบประลอง (tournament selection) เนื่องจากทำให้ LZWGA ทำงานได้มีประสิทธิภาพมากที่สุดเมื่อเทียบกับการคัดเลือกวิธีอื่น (อณิมา และ วรเศรษฐ, 2552) และใช้การไขว้เปลี่ยนแบบยูนิฟอร์มเพราะทำให้ LZWGA ทำงานได้มีประสิทธิภาพมากที่สุดเช่นกัน (สมรักษ์ และ วรเศรษฐ, 2550) การทดลองนี้จะไม่มีการกลายพันธุ์ ทำซ้ำการทดลองละ 10 ครั้ง พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad แสดงในตารางที่ 3 ถึงตารางที่ 6 ตามลำดับ

ตารางที่ 3 พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax

พารามิเตอร์	**ค่า**		
ความยาวของโครโมโซม LZW	50	150	1,000
ค่าความเหมาะสมของคำตอบ	1,000	10,000	100,000
จำนวนประชากร	100		
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว		
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม		
อัตราการไขว้เปลี่ยน	80%		
จำนวนรุ่นจำกัดไว้ที่	100	300	1,000

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 4 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomMax

พารามิเตอร์	ค่า	
ความยาวของโครโมโซม LZW	50	150
ค่าความเหมาะสมของคำตอบ	1,000	10,000
จำนวนประชากร	100	
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว	
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม	
อัตราการไขว้เปลี่ยน	80%	
จำนวนรุ่นจำกัดไว้ที่	100	200

ตารางที่ 5 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RoyalRoad

พารามิเตอร์	ค่า		
ความยาวของโครโมโซม LZW	50	150	1,000
ค่าความเหมาะสมของคำตอบ	1,000	10,000	100,000
จำนวนประชากร	100		
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว		
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม		
อัตราการไขว้เปลี่ยน	80%		
จำนวนรุ่นจำกัดไว้ที่	100	300	1,000
ขนาดบล็อก	10		

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 6 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad

พารามิเตอร์	ค่า	
ความยาวของโครโมโซม LZW	100	500
ค่าความเหมาะสมของคำตอบ	1,000	10,000
จำนวนประชากร	1,000	
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว	
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม	
อัตราการไขว้เปลี่ยน	80%	
จำนวนรุ่นจำกัดไว้ที่	500	
ขนาดบล็อก	4	

3.1.2 การทดลองเพื่อเปรียบเทียบความยาวโครโมโซมที่ใช้ในการพบคำตอบ

การทดลองเพื่อเปรียบเทียบความยาวโครโมโซมที่ใช้ในการพบคำตอบทำกับปัญหาทดสอบ 2 ปัญหา คือ OneMax และ RoyalRoad ซึ่งเป็นปัญหาที่ LZWGA สามารถหาคำตอบพบได้ โดยกำหนดขนาดปัญหาตั้งแต่ 1,000 บิตถึง 10,000 บิต ใช้อัตรา shift 0.5 (50%) พารามิเตอร์ที่ใช้ในการแก้ปัญหาทั้งสองแสดงในตารางที่ 7

ตารางที่ 7 พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax และ RoyalRoad เพื่อเทียบความยาว

พารามิเตอร์	ค่า
จำนวนประชากร	100
ค่าความเหมาะสมของคำตอบ	1,000 ถึง 10,000 บิต
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม
อัตราการไขว้เปลี่ยน	80%
อัตราการ shift	50%
ขนาดบล็อก	4

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3.1.3 การทดลองเพื่อเปรียบเทียบจำนวนรุ่นที่พบคำตอบ

การทดลองเพื่อเปรียบเทียบจำนวนรุ่นที่พบคำตอบทำกับปัญหาทดสอบ 2 ปัญหา คือ OneMax และ RoyalRoad เช่นเดียวกัน โดยกำหนดขนาดปัญหาตั้งแต่ 1,000 บิตถึง 10,000 บิต ใช้อัตรา shift 0.5 (50%) ทำซ้ำการทดลองละ 10 ครั้ง พารามิเตอร์ที่ใช้ในการแก้ปัญหาทั้งสองแสดงในตารางที่ 8

ตารางที่ 8 พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax และ RoyalRoad เพื่อเทียบจำนวนรุ่น

พารามิเตอร์	ค่า
จำนวนประชากร	100
ความยาวของโครโมโซม LZW	จากการทดลอง 3.1.2
ค่าความเหมาะสมของคำตอบ	1,000 ถึง 10,000 บิต
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม
อัตราการไขว้เปลี่ยน	80%
อัตราการกลายพันธุ์	0%
อัตราการ shift	50%
ขนาดบล็อก	4

3.2 การทดลองเพื่อวัดประสิทธิภาพของการปรับปรุงตัวดำเนินการ shift-and-check (Improved shift-and-check operator : ISC)

การทดลองในส่วนนี้กระทำกับ 3 ปัญหาทดสอบคือ OneMax,RoyalRoad, และ RandomRoyalRoad ใช้ 3 วิธีการคือ LZWGA ธรรมดา, LZWGA ที่ใช้ตัวดำเนินการ shift-and-check และ LZWGA ที่ใช้ตัวดำเนินการ ISC เปรียบเทียบประสิทธิภาพของโดยใช้ค่าความเหมาะสม หรือจำนวนรุ่นที่พบคำตอบ ทำซ้ำการทดลองละ 10 ครั้ง พารามิเตอร์ที่ใช้แสดงในตารางที่ 9 และ 10

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 9 พารามิเตอร์ที่ใช้ในการแก้ปัญหา OneMax และ RoyalRoad

พารามิเตอร์	ค่า
จำนวนประชากร	300
ความยาวของโครโมโซม LZW	1,000
ค่าความเหมาะสมของคำตอบ	100,000
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม
อัตราการไขว้เปลี่ยน	80%
อัตราการshift-and-check	50%
จำนวนรุ่นจำกัดไว้ที่	1,000
ขนาดบล็อก	10

ตารางที่ 10 พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad

พารามิเตอร์	ค่า
จำนวนประชากร	300
ความยาวของโครโมโซม LZW	2,000
ค่าความเหมาะสมของคำตอบ	100,000
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม
อัตราการไขว้เปลี่ยน	80%
อัตราการshift-and-check	50%
จำนวนรุ่นจำกัดไว้ที่	1,000
ขนาดบล็อก	4

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3.3 การทดลองเพื่อวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการ Shift และ Shift-and-Check ใน LZWGA

การทดลองเพื่อวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการ shift และ shift-and-check ใน LZWGA ทำกับปัญหาทดสอบทั้ง 4 ปัญหาเช่นเดียวกัน มีขั้นตอนดังนี้

(1) สร้างโครโมโซม LZWGA ทุกรูปแบบที่เป็นไปได้

(2) วัดค่าความเหมาะสมของโครโมโซมทุกตัว

(3) นำโครโมโซมแต่ละตัวมาผ่านการ shift ทีละตำแหน่ง เริ่มตั้งแต่ตำแหน่งแรกจนครบทุกตำแหน่งและวัดค่าความเหมาะสมที่ได้จากการ shift แต่ละครั้ง

(4) เปรียบเทียบค่าความเหมาะสมของโครโมโซมที่ได้หลังการ shift เทียบกับค่าความเหมาะสมของโครโมโซมต้นฉบับ

(5) คำนวณความน่าจะเป็นที่ shift ทำให้โครโมโซมของ LZWGA มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมโดยใช้สมการที่ 4

$$P = \frac{NC}{TC} \tag{4}$$

โดย

P คือ ความน่าจะเป็นที่ค่าความเหมาะสมของโครโมโซมเพิ่มขึ้นหรือเท่าเดิม

NC คือ จำนวนโครโมโซมที่เมื่อใช้ shift แล้วค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม

TC คือ จำนวนโครโมโซมทุกรูปแบบที่เป็นไปได้และนำมาผ่านการ shift ทุกตำแหน่ง ค่าของ TC คำนวณโดยใช้สมการที่ 5

$$TC = n \times (n+1)! \tag{5}$$

โดย n คือ ความยาวโครโมโซมของ LZWGA

การทดลองจะกำหนดความยาวโครโมโซมของ LZWGA ตั้งแต่ 4 ถึง 8 ตำแหน่ง หลังจากนั้นนำค่าความน่าจะเป็นที่การ shift จะทำให้ค่าความเหมาะสมของโครโมโซมของ LZWGA สูงขึ้นมาวาดเป็นกราฟ และสรุปผล

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ผลและวิจารณ์

ผล

1. ผลการทดลองเพื่อวัดประสิทธิภาพของตัวดำเนินการ Shift และ Shift-and-check

ผลการทดลองเพื่อวัดประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check แบ่งตามการเปรียบเทียบออกเป็น 3 กรณี คือ การเปรียบเทียบจากค่าความเหมาะสม, เปรียบเทียบจากความยาวโครโมโซมที่ใช้ในการพบคำตอบ, และเปรียบเทียบจำนวนรุ่นที่ใช้ในการพบคำตอบ

1.1 ผลจากการทดลองเพื่อเปรียบเทียบค่าความเหมาะสมของโครโมโซม

ผลการทดลองเพื่อเปรียบเทียบค่าความเหมาะสมของโครโมโซมที่ได้แบ่งตามปัญหาทดสอบทั้ง 4 ปัญหา คือ OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad

1.1.1 ปัญหา OneMax

ผลการทดลองกับปัญหา OneMax แบ่งตามขนาดโครโมโซมของ GA ออกเป็น 3 ขนาดคือ 1,000 บิต, 10,000 บิต และ 100,000 บิต เมื่อทดลองใช้ตัวดำเนินการ shift และ shift-and-check ด้วยอัตราตั้งแต่ 0.1 ถึง 1.0 พบว่าโครโมโซมที่ดีที่สุดมีค่าความเหมาะสมดังแสดงในตารางที่ 11, 12 และ 13 ตามลำดับ จะเห็นว่าการ shift-and-check ช่วยให้ LZWGA สามารถหาคำตอบพบโดยมีจำนวนรุ่นเฉลี่ยในการพบคำตอบและโอกาสในการพบคำตอบแสดงดังท้ายตาราง ซึ่งอัตราการ shift-and-check นั้นส่วนใหญ่จะแปรผกผันต่อจำนวนรุ่นที่พบคำตอบ นั่นคือ ถ้าอัตราการเกิดการ shift-and-check มาก ก็จะทำให้จำนวนรุ่นที่ใช้ในการพบคำตอบน้อยลง หรือหาคำตอบพบได้เร็วขึ้นนั่นเอง แต่ถึงแม้อัตราการ shift-and-check สูง จะทำให้พบคำตอบเร็วขึ้นในบางกรณี แต่ในบางกรณีก็มีโอกาสที่จะคลาดเคลื่อนและหาคำตอบพบน้อยลงด้วยเช่นกัน ส่วนตัวดำเนินการ shift ถึงแม้ในกรณีนี้จะหาคำตอบไม่พบ (เนื่องจากกลุ่มประชากรมีขนาดเล็ก) แต่ก็ช่วยทำให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมสูงกว่า LZWGA ธรรมดาเสมอ

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 11 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆในปัญหา OneMax ขนาด 1,000 บิต

อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift	shift-and-check		
			ค่าความเหมาะสม	รุ่นที่พบ	พบคำตอบ (%)
0.1	480.6	526.9	1,000.0	40.7	100
0.2	480.6	544.3	1,000.0	35.2	100
0.3	480.6	618.9	1,000.0	32.6	100
0.4	480.6	590.2	1,000.0	26.5	100
0.5	480.6	596.3	1,000.0	26.2	100
0.6	480.6	616.9	998.0	25.1	80
0.7	480.6	610.6	999.5	22.5	90
0.8	480.6	574.9	999.6	20.6	90
0.9	480.6	579.6	1,000.0	20.5	100
1.0	480.6	570.5	998.2	20.1	90

ตารางที่ 12 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆในปัญหา OneMax ขนาด 10,000บิต

อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift	shift-and-check		
			ค่าความเหมาะสม	รุ่นที่พบ	พบคำตอบ (%)
0.1	1,871.4	2,012.3	7,908.4	263.0	70
0.2	1,871.4	2,066.7	9,641.1	218.5	80
0.3	1,871.4	2,335.6	10,000.0	252.7	100
0.4	1,871.4	2,333.7	9,653.3	152.5	40
0.5	1,871.4	2,458.9	9,582.0	197.7	40
0.6	1,871.4	2,467.3	9,619.2	121.2	50
0.7	1,871.4	2,410.3	9,368.1	112.2	40
0.8	1,871.4	2,453.5	8,839.4	104.7	40
0.9	1,871.4	2,629.0	8,593.9	84.5	30
1.0	1,871.4	2,498.6	8,675.7	76.0	20

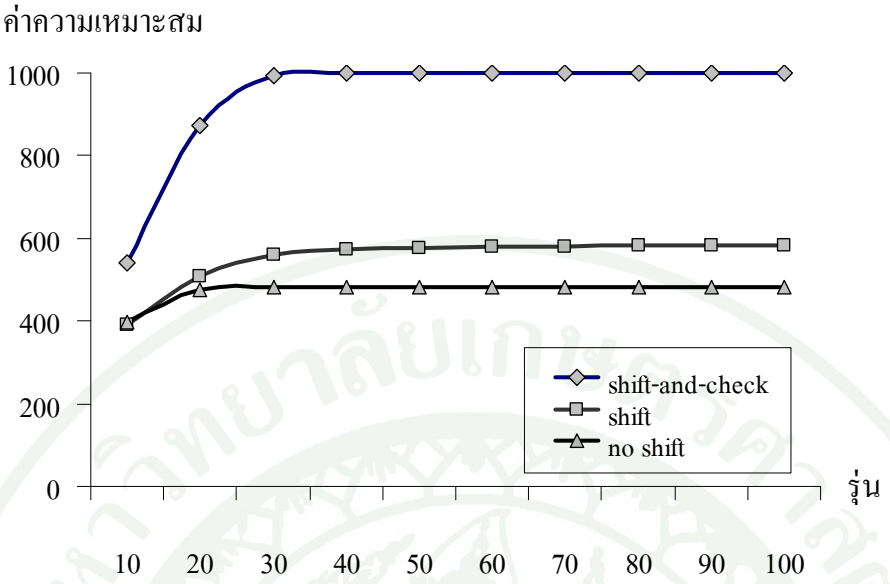
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 13 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา OneMax ขนาด 100,000 บิต

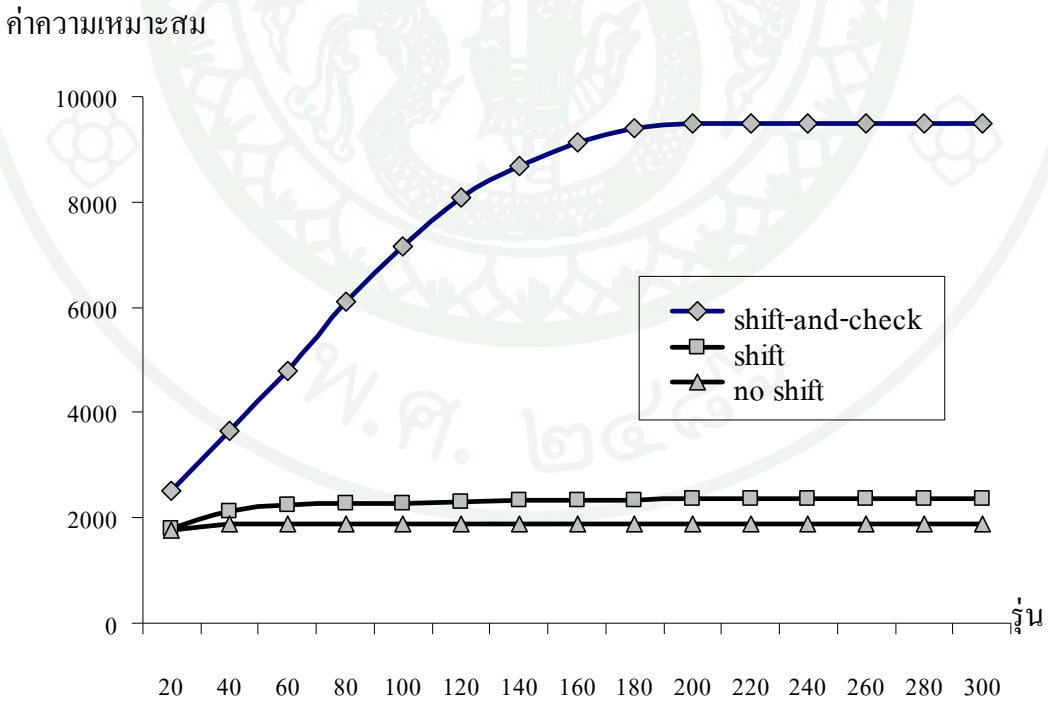
อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift	shift-and-check		
			ค่าความเหมาะสม	รุ่นที่พบ	พบคำตอบ (%)
0.1	15,703.6	17,244.4	62,553.2	ไม่พบ	0
0.2	15,703.6	17,208.2	63,934.6	ไม่พบ	0
0.3	15,703.6	19,313.4	91,333.2	ไม่พบ	0
0.4	15,703.6	19,940.4	85,506.8	864.0	20
0.5	15,703.6	20,528.0	98,580.0	817.5	40
0.6	15,703.6	22,820.4	97,827.4	862.2	80
0.7	15,703.6	22,826.2	95,676.4	863.0	80
0.8	15,703.6	22,637.6	92,742.8	761.0	80
0.9	15,703.6	23,305.0	98,490.0	712.0	80
1.0	15,703.6	21,806.8	100,000.0	639.2	100

ค่าความเหมาะสมของโครโมโซมที่ได้จากการทดลองทั้ง 3 วิธีการจากปัญหาทั้ง 3 ขนาดในแต่ละรุ่นของวิวัฒนาการโดยใช้อัตราการ shift เท่ากับ 0.5 แสดงในภาพที่ 13, 14 และ 15 จากภาพจะเห็นว่า เมื่อ LZWGA ธรรมดาหยุดวิวัฒนาการแล้ว แต่ LZWGA ที่ใช้ตัวดำเนินการ shift และ shift-and-check ยังช่วยให้ LZWGA สร้างโครโมโซมที่มีความเหมาะสมสูงขึ้นต่อไปอีก โดย LZWGA ที่ใช้ตัวดำเนินการ shift-and-check จะมีวิวัฒนาการจนสามารถพบคำตอบได้

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

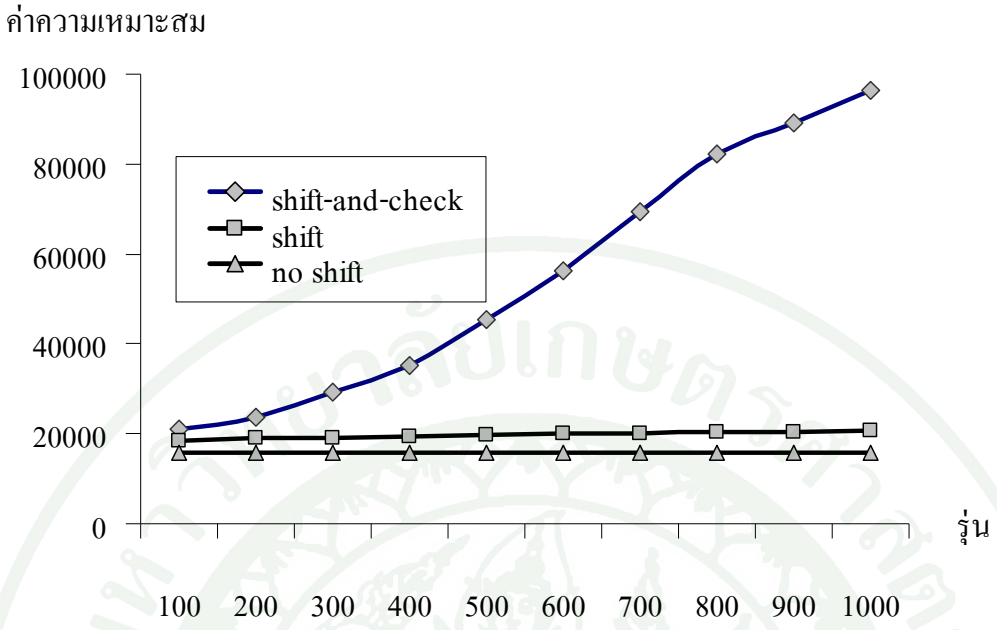


ภาพที่ 13 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ check ในปัญหา OneMax ขนาด 1,000 บิต



ภาพที่ 14 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ check ในปัญหา OneMax ขนาด 10,000 บิต

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 15 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ check ในปัญหา OneMax ขนาด 100,000 บิต

การเพิ่มตัวดำเนินการ shift และ shift-and-check เข้ามานั้นทำ LZWGA มีความยุ่งยากและเสียเวลามากขึ้น แต่ในขณะเดียวกันก็ทำให้ได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นตามไปด้วย เวลาที่ใช้ทั้งหมดในการค้นหาคำตอบจากปัญหา OneMax ที่อัตราการ shift ระดับต่างๆ แสดงในตารางที่ 14 จากตารางจะเห็นว่าตัวดำเนินการ shift ทำให้ LZWGA ใช้เวลาในการทำงานมากขึ้น แต่ช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น ในขณะที่ตัวดำเนินการ shift-and-check ถึงแม้จะมีขั้นตอนที่ยุ่งยากมากกว่า แต่กลับช่วยลดเวลาในการทำงานลง เพราะเมื่อพบคำตอบแล้ว LZWGA จะหยุดการทำงานไม่ต้องวนรอบไปจนถึงจำนวนรุ่นสูงสุดที่กำหนดไว้

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 14 เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา OneMax ขนาด 1,000 บิต

วิธีการ	เวลาเฉลี่ยที่ใช้ (วินาที)	เงื่อนไขในการหยุดการทำงาน	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด
No shift	0.96	ถึงจำนวนรุ่นที่จำกัดไว้	480.6
shift 0.1	1.08	ถึงจำนวนรุ่นที่จำกัดไว้	526.9
shift-and-check 0.1	0.16	พบคำตอบ	1,000.0
shift 0.2	1.19	ถึงจำนวนรุ่นที่จำกัดไว้	544.3
shift-and-check 0.2	0.12	พบคำตอบ	1,000.0
shift 0.3	1.20	ถึงจำนวนรุ่นที่จำกัดไว้	618.9
shift-and-check 0.3	0.11	พบคำตอบ	1,000.0
shift 0.4	1.27	ถึงจำนวนรุ่นที่จำกัดไว้	590.2
shift-and-check 0.4	0.10	พบคำตอบ	1,000.0
shift 0.5	1.40	ถึงจำนวนรุ่นที่จำกัดไว้	596.3
shift-and-check 0.5	0.09	พบคำตอบ	1,000.0
shift 0.6	1.45	ถึงจำนวนรุ่นที่จำกัดไว้	616.9
shift-and-check 0.6	0.12	พบคำตอบ	1,000.0
shift 0.7	1.45	ถึงจำนวนรุ่นที่จำกัดไว้	610.6
shift-and-check 0.7	0.11	พบคำตอบ	1,000.0
shift 0.8	1.48	ถึงจำนวนรุ่นที่จำกัดไว้	574.9
shift-and-check 0.8	0.09	พบคำตอบ	1,000.0
shift 0.9	1.58	ถึงจำนวนรุ่นที่จำกัดไว้	579.6
shift-and-check 0.9	0.12	พบคำตอบ	1,000.0
shift 1.0	1.62	ถึงจำนวนรุ่นที่จำกัดไว้	570.5
shift-and-check 1.0	0.14	พบคำตอบ	1,000.0

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

1.2 ปัญหา RandomMax

การทดลองกับปัญหา RandomMax แบ่งตามขนาดโครโมโซมของ GA ออกเป็น 2 ขนาดคือ 1,000 บิต และ 10,000 บิต เมื่อทดลองใช้ตัวดำเนินการ shift และ shift-and-check ด้วยอัตราตั้งแต่ 0.1 ถึง 1.0 พบว่าโครโมโซมที่ดีที่สุดมีค่าความเหมาะสมดังแสดงในตารางที่ 15 และ 16 ตามลำดับ จะเห็นว่าการ shift และ shift-and-check ช่วยให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น โดยตัวดำเนินการ shift ช่วยให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมเฉลี่ยสูงกว่า LZWGA ธรรมดา 1.20 เท่า ในขณะที่ ตัวดำเนินการ shift-and-check ช่วยให้ LZWFA สร้างโครโมโซมที่มีค่าวามเหมาะสมเฉลี่ยสูงกว่า LZWGA ธรรมดา 3.12 เท่า

จากตารางจะเห็นว่า อัตราการ shift มาก ไม่ได้ทำให้โครโมโซมมีค่าความเหมาะสมเฉลี่ยสูงขึ้นเสมอไป ดังนั้นในการนำไปใช้ควรใช้อัตราการ shift ที่ระดับกลางคือประมาณ 0.5 จึงจะเหมาะสมที่สุด

ตารางที่ 15 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RandomMax ขนาด 1,000 บิต

อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift		shift-and-check	
		ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)	ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)
0.1	268.9	295.0	1.10	512.4	1.91
0.2	268.9	302.8	1.13	503.7	1.87
0.3	268.9	305.6	1.14	507.7	1.89
0.4	268.9	313.5	1.17	506.5	1.88
0.5	268.9	321.3	1.19	510.7	1.90
0.6	268.9	319.3	1.19	519.1	1.93
0.7	268.9	327.5	1.22	510.0	1.90
0.8	268.9	331.2	1.23	508.9	1.89
0.9	268.9	314.0	1.17	507.9	1.89
1.0	268.9	318.8	1.19	507.9	1.89

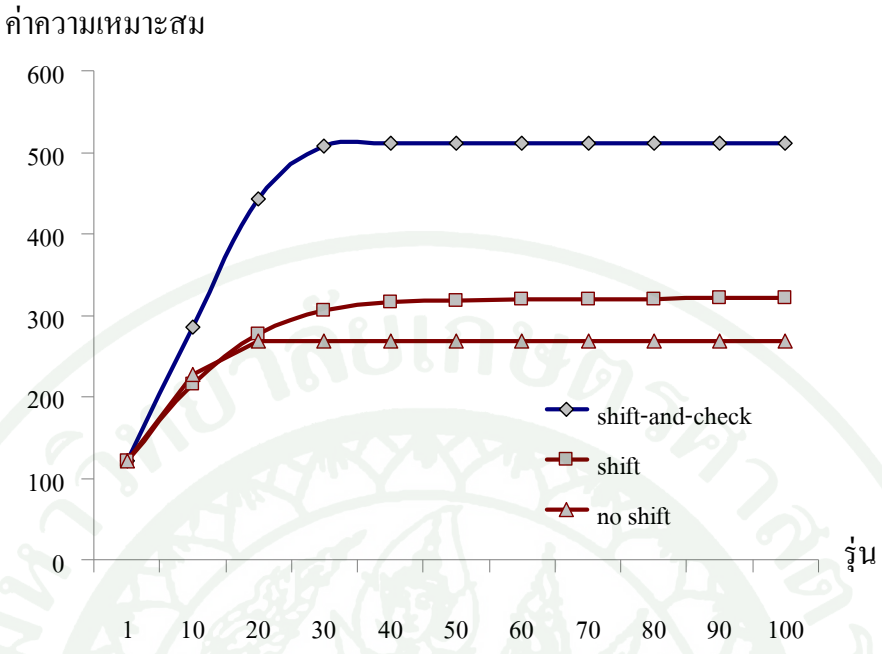
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 16 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RandomMax ขนาด 10,000 บิต

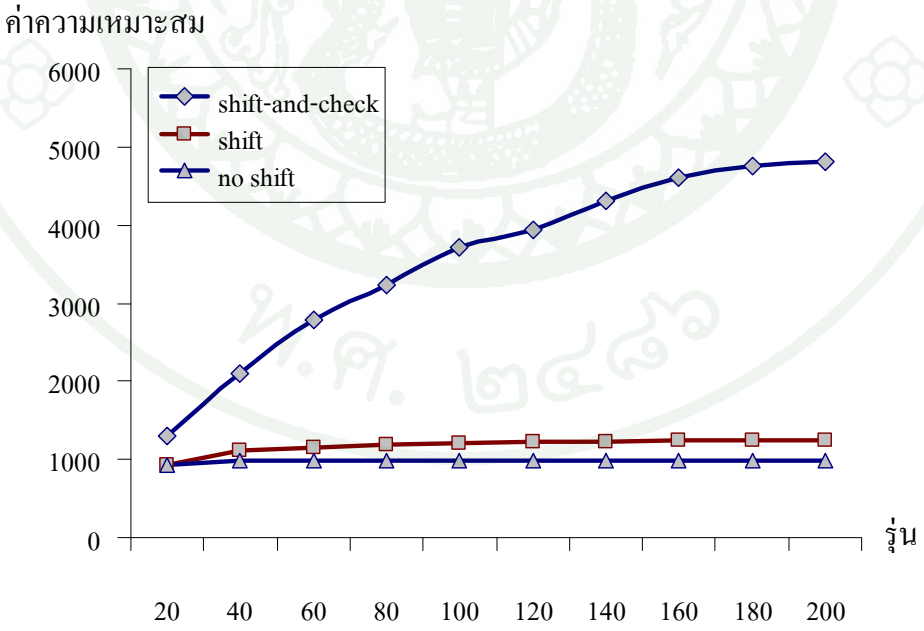
อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift		shift-and-check	
		ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)	ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)
0.1	981.3	1,053.1	1.07	2,193.7	2.24
0.2	981.3	1,059.3	1.08	4,448.9	4.53
0.3	981.3	1,206.8	1.23	3,326.5	3.39
0.4	981.3	1,231.4	1.25	4,494.9	4.58
0.5	981.3	1,247.8	1.27	4,885.6	4.98
0.6	981.3	1,246.5	1.27	4,652.4	4.74
0.7	981.3	1,223.2	1.25	4,614.3	4.70
0.8	981.3	1,265.9	1.29	4,507.3	4.59
0.9	981.3	1,271.4	1.30	4,784.6	4.88
1.0	981.3	1,346.2	1.37	4,747.0	4.84

ค่าความเหมาะสมของโครโมโซมที่ได้จากการทดลองทั้ง 3 วิธีการจากปัญหาทั้ง 2 ขนาดในแต่ละรุ่นของวิวัฒนาการโดยใช้อัตราการ shift เท่ากับ 0.5 แสดงในภาพที่ 16 และ 17 จากภาพจะเห็นว่า เมื่อ LZWGA ธรรมดาหยุดวิวัฒนาการแล้ว แต่ LZWGA ที่ใช้ตัวดำเนินการ shift และ shift-and-check ยังช่วยให้ LZWGA สร้างโครโมโซมที่มีความเหมาะสมสูงขึ้นต่อไปอีก

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 16 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomMax ขนาด 1,000 บิต



ภาพที่ 17 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomMax ขนาด 10,000 บิต

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สำหรับในปัญหา RandomMax ค่าของโครโมโซมที่เป็นคำตอบเกิดจากการสุ่ม การเพิ่มตัวดำเนินการ shift และ shift-and-check เข้ามานั้นทำ LZWGA มีความยุ่งยากและเสียเวลามากขึ้นเช่นเดียวกับในปัญหา OneMax แต่ในขณะเดียวกันก็ทำให้ได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นตามไปด้วย เวลาที่ใช้ทั้งหมดในการค้นหาคำตอบจากปัญหา RandomMax ที่อัตราการ shift ระดับต่างๆ แสดงในตารางที่ 17 จะเห็นว่าทั้งตัวดำเนินการ shift และ shift-and-check ต่างก็ทำให้ LZWGA เสียเวลามากขึ้น แต่ช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น

จากตารางจะเห็นว่าบางครั้งอัตราการ shift มากแต่ใช้เวลาในการทำงานน้อยกว่าอัตราการ shift ที่น้อย เป็นเพราะว่าเวลารวมที่ใช้มิได้ขึ้นอยู่กับอัตราการ shift เพียงอย่างเดียว หากแต่ยังขึ้นอยู่กับตำแหน่งที่เกิดการ shift ด้วย ตำแหน่งที่ถูกสุ่มให้เกิดการ shift ถ้าเป็นตำแหน่งหลังๆ ย่อมจะเสียเวลาน้อยกว่าตำแหน่งต้นๆ ของโครโมโซม

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 17 เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา RandomMax ขนาด 1,000 บิต

วิธีการ	เวลาเฉลี่ยที่ใช้ (วินาที)	เงื่อนไขในการหยุดการทำงาน	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด
No shift	11.77	ถึงจำนวนรุ่นที่จำกัดไว้	505.0
shift 0.1	11.79	ถึงจำนวนรุ่นที่จำกัดไว้	533.6
shift-and-check 0.1	12.20	ถึงจำนวนรุ่นที่จำกัดไว้	553.2
shift 0.2	12.88	ถึงจำนวนรุ่นที่จำกัดไว้	530.8
shift-and-check 0.2	13.20	ถึงจำนวนรุ่นที่จำกัดไว้	555.8
shift 0.3	13.24	ถึงจำนวนรุ่นที่จำกัดไว้	543.4
shift-and-check 0.3	14.17	ถึงจำนวนรุ่นที่จำกัดไว้	554.6
shift 0.4	14.42	ถึงจำนวนรุ่นที่จำกัดไว้	538.0
shift-and-check 0.4	15.25	ถึงจำนวนรุ่นที่จำกัดไว้	561.0
shift 0.5	13.85	ถึงจำนวนรุ่นที่จำกัดไว้	539.2
shift-and-check 0.5	15.18	ถึงจำนวนรุ่นที่จำกัดไว้	559.4
shift 0.6	13.67	ถึงจำนวนรุ่นที่จำกัดไว้	539.2
shift-and-check 0.6	14.62	ถึงจำนวนรุ่นที่จำกัดไว้	559.4
shift 0.7	15.09	ถึงจำนวนรุ่นที่จำกัดไว้	536.8
shift-and-check 0.7	14.28	ถึงจำนวนรุ่นที่จำกัดไว้	563.4
shift 0.8	14.10	ถึงจำนวนรุ่นที่จำกัดไว้	531.8
shift-and-check 0.8	14.62	ถึงจำนวนรุ่นที่จำกัดไว้	557.6
shift 0.9	13.40	ถึงจำนวนรุ่นที่จำกัดไว้	537.4
shift-and-check 0.9	14.47	ถึงจำนวนรุ่นที่จำกัดไว้	554.6
shift 1.0	14.44	ถึงจำนวนรุ่นที่จำกัดไว้	539.0
shift-and-check 1.0	15.40	ถึงจำนวนรุ่นที่จำกัดไว้	564.2

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

1.3 ปัญหา RoyalRoad

ผลการทดลองกับปัญหา RoyalRoad แบ่งตามขนาดโครโมโซมของ GA ออกเป็น 3 ขนาดคือ 1,000 บิต, 10,000 บิต และ 100,000 บิต เมื่อทดลองใช้ตัวดำเนินการ shift และ shift-and-check ด้วยอัตราตั้งแต่ 0.1 ถึง 1.0 พบว่าโครโมโซมที่ดีที่สุดมีค่าความเหมาะสมดังแสดงในตารางที่ 18,19 และ 20 ตามลำดับ จะเห็นว่าการ shift-and-check ช่วยให้ LZWGA สามารถหาคำตอบพบ โดยมีจำนวนรุ่นเฉลี่ยในการพบคำตอบและโอกาสในการพบคำตอบแสดงดังท้ายตาราง ซึ่งอัตราการ shift-and-check นั้นส่วนใหญ่จะแปรผกผันต่อจำนวนรุ่นที่พบคำตอบ นั่นคือ ถ้าอัตราการเกิดการ shift-and-check มาก ก็จะทำให้จำนวนรุ่นที่ใช้ในการพบคำตอบน้อยลง หรือหาคำตอบพบได้เร็วขึ้นนั่นเอง แต่ถึงแม้อัตราการ shift สูง จะทำให้พบคำตอบเร็วขึ้นในบางกรณี แต่ในบางกรณีก็มีโอกาสที่จะคลาดเคลื่อนและหาคำตอบพบน้อยลงด้วยเช่นกัน ส่วนตัวดำเนินการ shift ถึงแม้ในกรณีนี้จะหาคำตอบไม่พบ(เนื่องจากกลุ่มประชากรมีขนาดเล็ก) แต่ก็ช่วยทำให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมสูงกว่า LZWGA ธรรมดาเสมอ

ตารางที่ 18 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆในปัญหา RoyalRoad ขนาด 1,000 บิต

อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift	shift-and-check		
			ค่าความเหมาะสม	รุ่นที่พบ	พบคำตอบ (%)
0.1	460.0	496.0	1,000.0	44.2	100
0.2	460.0	548.0	1,000.0	33.8	100
0.3	460.0	545.0	1,000.0	29.2	100
0.4	460.0	533.0	1,000.0	28.7	100
0.5	460.0	534.0	1,000.0	24.4	100
0.6	460.0	579.0	1,000.0	25.5	100
0.7	460.0	552.0	1,000.0	25.8	100
0.8	460.0	540.0	991.0	21.7	90
0.9	460.0	520.0	988.0	19.7	90
1.0	460.0	519.0	993.0	22.8	90

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 19 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆในปัญหา OneMax ขนาด 10,000 บิต

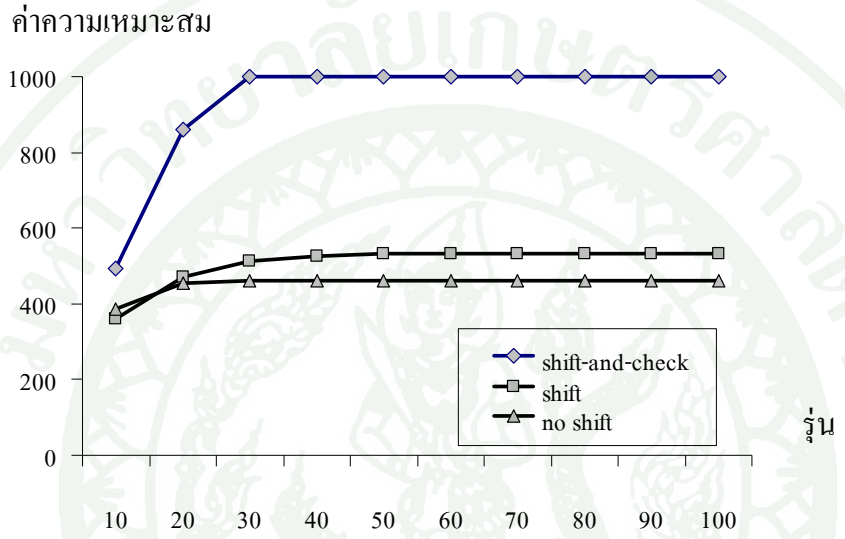
อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift	shift-and-check		
			ค่าความเหมาะสม	รุ่นที่พบ	พบคำตอบ (%)
0.1	1,769.0	2,040.0	8,286.0	297.8	70
0.2	1,769.0	2,058.0	9,324.0	214.1	90
0.3	1,769.0	2,254.0	9,445.0	291.0	30
0.4	1,769.0	2,176.0	9,957.0	161.8	80
0.5	1,769.0	2,209.0	9,727.0	146.8	50
0.6	1,769.0	2,324.0	9,415.0	108.5	40
0.7	1,769.0	2,350.0	9,132.0	106.3	30
0.8	1,769.0	2,411.0	9,065.0	94.0	20
0.9	1,769.0	2,344.0	9,180.0	96.7	40
1.0	1,769.0	2,238.0	8,395.0	80.3	30

ตารางที่ 20 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RoyalRoad ขนาด 100,000 บิต

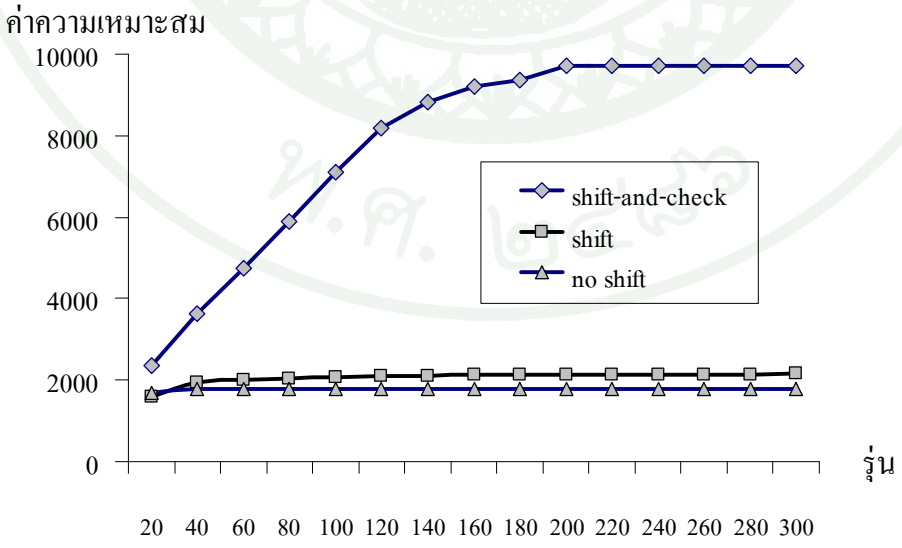
อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift	shift-and-check		
			ค่าความเหมาะสม	รุ่นที่พบ	พบคำตอบ (%)
0.1	14,493.0	15,694.0	42,746.0	ไม่พบ	0
0.2	14,493.0	15,490.0	54,292.0	ไม่พบ	0
0.3	14,493.0	16,788.0	80,848.0	734.0	40
0.4	14,493.0	17,758.0	90,486.0	912.0	40
0.5	14,493.0	19,642.0	99,998.0	883.2	60
0.6	14,493.0	21,422.0	91,000.0	800.6	60
0.7	14,493.0	20,278.0	96,286.0	623.0	60
0.8	14,493.0	20,576.0	95,872.0	734.5	80
0.9	14,493.0	22,162.0	88,972.0	626.0	40
1.0	14,493.0	21,816.0	98,306.0	532.7	80

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ค่าความเหมาะสมของโครโมโซมที่ได้จากการทดลองทั้ง 3 วิธีการจากปัญหาทั้ง 3 ขนาดในแต่ละรุ่นของวิวัฒนาการโดยใช้อัตราการ shift เท่ากับ 0.5 แสดงในภาพที่ 18,19 และ 20 จากภาพจะเห็นว่า เมื่อ LZWGA ธรรมดาหยุดวิวัฒนาการแล้ว แต่ LZWGA ที่ใช้ตัวดำเนินการ shift และ shift-and-check ยังช่วยให้ LZWGA สร้างโครโมโซมที่มีความเหมาะสมสูงขึ้นต่อไปอีก โดย LZWGA ที่ใช้ตัวดำเนินการ shift-and-check จะมีวิวัฒนาการจนสามารถพบคำตอบได้

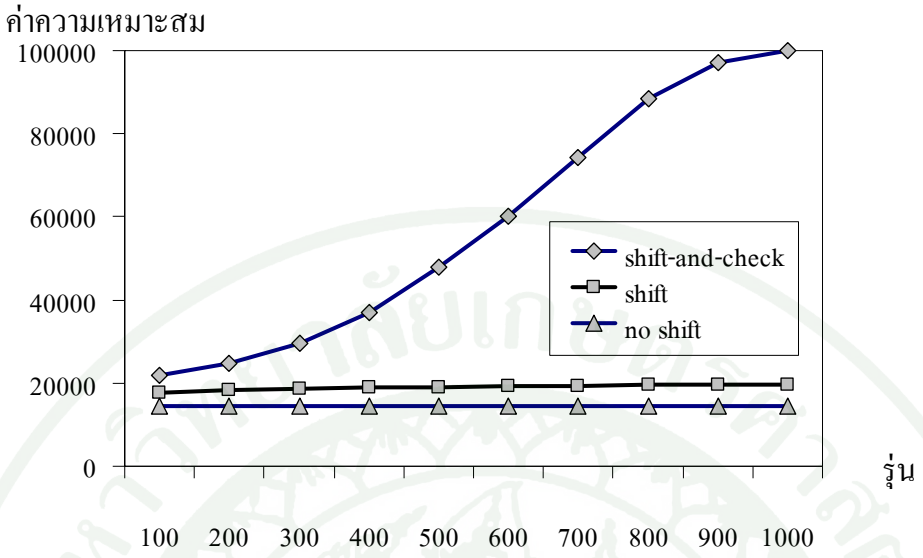


ภาพที่ 18 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ check ในปัญหา RoyalRoad ขนาด 1,000 บิต



ภาพที่ 19 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ check ในปัญหา OneMax ขนาด 10,000 บิต

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 20 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ check ในปัญหา RoyalRoad ขนาด 100,000 บิต

สำหรับปัญหา RoyalRoad การเพิ่มตัวดำเนินการ shift และ shift-and-check เข้ามานั้น ทำ LZWGA มีความยุ่งยากและเสียเวลาเพิ่มขึ้นเช่นเดียวกับในปัญหาอื่นๆ แต่ในขณะเดียวกันก็ทำให้ได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นตามไปด้วย เวลาที่ใช้ทั้งหมดในการค้นหาคำตอบจากปัญหา RoyalRoad ที่อัตราการ shift ระดับต่างๆ แสดงในตารางที่ 21 จากตารางจะเห็นว่าตัวดำเนินการ shift ทำให้ LZWGA เสียเวลามากขึ้น แต่ช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น ในขณะที่ตัวดำเนินการ shift-and-check ถึงแม้จะมีขั้นตอนที่ยุ่งยากมากกว่า แต่กลับช่วยลดเวลาในการค้นหาคำตอบลง เพราะเมื่อพบคำตอบแล้ว LZWGA จะหยุดการทำงานไม่ต้องวนรอบไปจนถึงจำนวนรุ่นสูงสุดที่กำหนดไว้

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 21 เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา RoyalRoad ขนาด 1,000 บิต

วิธีการ	เวลาเฉลี่ยที่ใช้ (วินาที)	เงื่อนไขในการหยุดการทำงาน	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด
No shift	0.92	ถึงจำนวนรุ่นที่จำกัดไว้	460.0
shift 0.1	0.97	ถึงจำนวนรุ่นที่จำกัดไว้	496.0
shift-and-check 0.1	0.35	พบคำตอบ	1,000.0
shift 0.2	1.48	ถึงจำนวนรุ่นที่จำกัดไว้	548.0
shift-and-check 0.2	0.44	พบคำตอบ	1,000.0
shift 0.3	1.52	ถึงจำนวนรุ่นที่จำกัดไว้	545.0
shift-and-check 0.3	0.37	พบคำตอบ	1,000.0
shift 0.4	1.78	ถึงจำนวนรุ่นที่จำกัดไว้	533.0
shift-and-check 0.4	0.36	พบคำตอบ	1,000.0
shift 0.5	1.99	ถึงจำนวนรุ่นที่จำกัดไว้	534.0
shift-and-check 0.5	0.34	พบคำตอบ	1,000.0
shift 0.6	2.07	ถึงจำนวนรุ่นที่จำกัดไว้	579.0
shift-and-check 0.6	0.31	พบคำตอบ	1,000.0
shift 0.7	2.07	ถึงจำนวนรุ่นที่จำกัดไว้	552.0
shift-and-check 0.7	0.49	พบคำตอบ	1,000.0
shift 0.8	2.08	ถึงจำนวนรุ่นที่จำกัดไว้	540.0
shift-and-check 0.8	0.42	พบคำตอบ	1,000.0
shift 0.9	2.09	ถึงจำนวนรุ่นที่จำกัดไว้	520.0
shift-and-check 0.9	0.49	พบคำตอบ	1,000.0
shift 1.0	2.11	ถึงจำนวนรุ่นที่จำกัดไว้	519.0
shift-and-check 1.0	0.50	พบคำตอบ	1,000.0

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

1.4 ผลการทดลองกับปัญหา RandomRoyalRoad

ผลการทดลองกับปัญหา RandomRoyalRoad แบ่งตามขนาดโครโมโซมของ GA ออกเป็น 2 ขนาดคือ 1,000 บิต และ 10,000 บิต เมื่อทดลองใช้ตัวดำเนินการ shift และ shift-and-check ด้วยอัตราตั้งแต่ 0.1 ถึง 1.0 พบว่าโครโมโซมที่ดีที่สุดมีค่าความเหมาะสมดังแสดงในตารางที่ 22 และ 23 ตามลำดับ จะเห็นว่าการ shift และ shift-and-check ช่วยให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น โดยตัวดำเนินการ shift ช่วยให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมเฉลี่ยสูงกว่า LZWGA ธรรมดา 1.45 เท่า ในขณะที่ ตัวดำเนินการ shift-and-check ช่วยให้ LZWFA สร้างโครโมโซมที่มีค่าวามเหมาะสมเฉลี่ยสูงกว่า LZWGA ธรรมดา 1.79 เท่า

จากตารางจะเห็นว่า อัตราการ shift มาก ไม่ได้ทำให้โครโมโซมมีค่าความเหมาะสมเฉลี่ยสูงขึ้นเสมอไป ดังนั้นในการนำไปใช้ควรใช้อัตราการ shift ที่ระดับกลางคือประมาณ 0.5 จึงจะเหมาะสมที่สุด

ตารางที่ 22 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RandomRoyalRoad ขนาด 1,000 บิต

อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift		shift-and-check	
		ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)	ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)
0.1	228.8	315.2	1.38	485.2	2.12
0.2	228.8	351.6	1.54	502.1	2.19
0.3	228.8	413.2	1.81	522.6	2.28
0.4	228.8	397.5	1.74	534.4	2.34
0.5	228.8	420.8	1.84	566.4	2.48
0.6	228.8	430.9	1.88	575.0	2.51
0.7	228.8	441.6	1.93	548.2	2.40
0.8	228.8	427.5	1.87	560.8	2.45
0.9	228.8	440.2	1.92	602.7	2.63
1.0	228.8	438.6	1.92	610.8	2.67

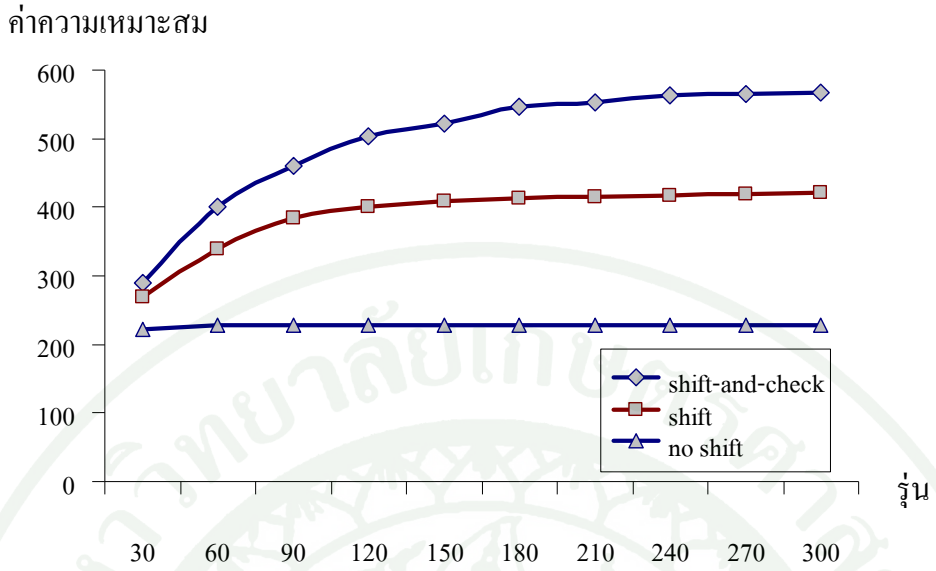
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 23 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด ที่อัตราการ shift ระดับต่างๆ ในปัญหา RandomRoyalRoad ขนาด 10,000 บิต

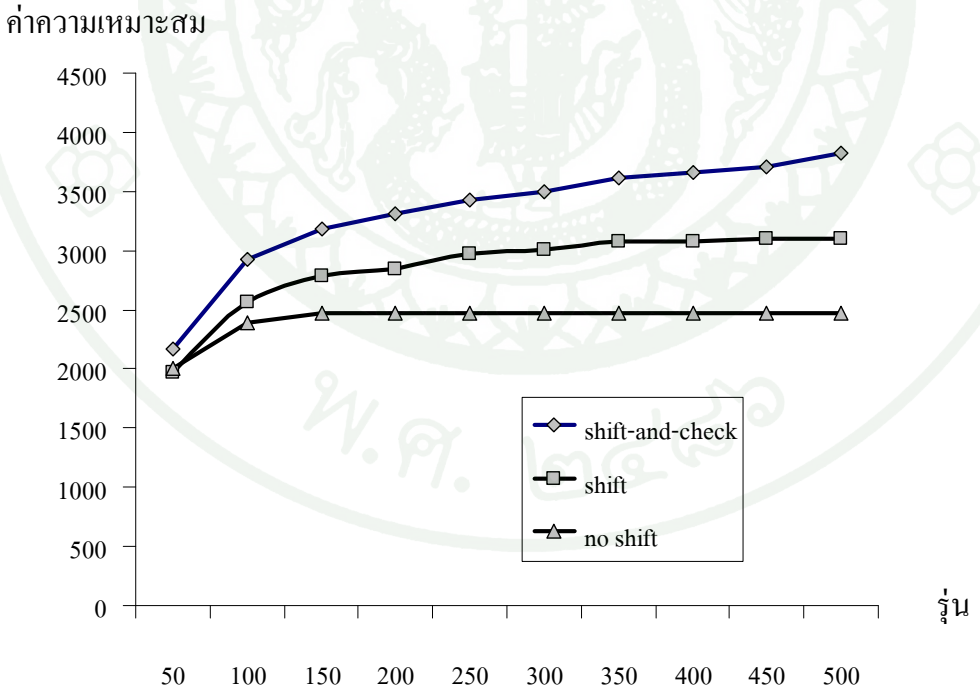
อัตราการ shift	วิธีการทดลอง				
	ไม่ shift	shift		shift-and-check	
		ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)	ค่าความเหมาะสมเฉลี่ย	เพิ่มขึ้น (เท่า)
0.1	2,468.0	2,813.9	1.14	3,601.2	1.46
0.2	2,468.0	3,012.6	1.22	3,702.4	1.50
0.3	2,468.0	3,108.7	1.26	3,402.5	1.38
0.4	2,468.0	3,215.3	1.30	3,853.3	1.56
0.5	2,468.0	3,104.2	1.26	3,820.5	1.55
0.6	2,468.0	3,208.9	1.30	3,904.8	1.58
0.7	2,468.0	3,301.4	1.34	4,002.6	1.62
0.8	2,468.0	3,238.6	1.31	3,976.5	1.61
0.9	2,468.0	3,189.7	1.29	3,990.8	1.62
1.0	2,468.0	3,403.2	1.38	3,986.4	1.62

ค่าความเหมาะสมของโครโมโซมที่ได้จากการทดลองทั้ง 3 วิธีการจากปัญหาทั้ง 2 ขนาดในแต่ละรุ่นของวิวัฒนาการโดยใช้อัตราการ shift เท่ากับ 0.5 แสดงในภาพที่ 21 และ 22 จากภาพจะเห็นว่า เมื่อ LZWGA ธรรมดาหยุดวิวัฒนาการแล้ว แต่ LZWGA ที่ใช้ตัวดำเนินการ shift และ shift-and-check ยังช่วยให้ LZWGA สร้างโครโมโซมที่มีความเหมาะสมสูงขึ้นต่อไปอีก

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 21 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomRoyalRoad ขนาด 1,000 บิต



ภาพที่ 22 ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift-and-check ในปัญหา RandomRoyalRoad ขนาด 10,000 บิต

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สำหรับในปัญหา RandomRoyalRoad ค่าของโครโมโซมที่เป็นบล็อกคำตอบเกิดจากการสุ่ม การเพิ่มตัวดำเนินการ shift และ shift-and-check เข้ามานั้นทำ LZWGA มีความยุ่งยากและเสียเวลามากขึ้นเช่นเดียวกับในปัญหาอื่นๆ แต่ในขณะเดียวกันก็ทำให้ได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นตามไปด้วย เวลาที่ใช้ทั้งหมดในการค้นหาคำตอบที่อัตราการ shift ระดับต่างๆ แสดงในตารางที่ 24 จะเห็นว่าทั้งตัวดำเนินการ shift และ shift-and-check ต่างก็ทำให้ LZWGA เสียเวลามากขึ้น แต่ช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น

จากตารางจะเห็นว่าบางครั้งอัตราการ shift มากแต่ใช้เวลาในการทำงานน้อยกว่าอัตราการ shift ที่น้อย เป็นเพราะว่าเวลารวมที่ใช้มิได้ขึ้นอยู่กับอัตราการ shift เพียงอย่างเดียว หากแต่ยังขึ้นอยู่กับตำแหน่งที่เกิดการ shift ด้วย ตำแหน่งที่ถูกสุ่มให้เกิดการ shift ถ้าเป็นตำแหน่งหลังๆ ย่อมจะเสียเวลาน้อยกว่าตำแหน่งต้นๆ ของโครโมโซม

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 24 เวลารวมของระบบที่ LZWGA ใช้ในการค้นหาคำตอบ ปัญหา RandomRoyalRoad ขนาด 1,000 บิต

วิธีการ	เวลาเฉลี่ยที่ใช้ (วินาที)	เงื่อนไขในการหยุดการทำงาน	ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด
No shift	2.18	ถึงจำนวนรุ่นที่จำกัดไว้	241.2
shift 0.1	2.18	ถึงจำนวนรุ่นที่จำกัดไว้	278.4
shift-and-check 0.1	2.62	ถึงจำนวนรุ่นที่จำกัดไว้	337.6
shift 0.2	2.55	ถึงจำนวนรุ่นที่จำกัดไว้	346.4
shift-and-check 0.2	2.60	ถึงจำนวนรุ่นที่จำกัดไว้	376.0
shift 0.3	2.39	ถึงจำนวนรุ่นที่จำกัดไว้	373.2
shift-and-check 0.3	2.61	ถึงจำนวนรุ่นที่จำกัดไว้	434.4
shift 0.4	2.46	ถึงจำนวนรุ่นที่จำกัดไว้	385.6
shift-and-check 0.4	2.61	ถึงจำนวนรุ่นที่จำกัดไว้	500.8
shift 0.5	2.50	ถึงจำนวนรุ่นที่จำกัดไว้	442.0
shift-and-check 0.5	2.61	ถึงจำนวนรุ่นที่จำกัดไว้	593.6
shift 0.6	2.62	ถึงจำนวนรุ่นที่จำกัดไว้	535.2
shift-and-check 0.6	2.63	ถึงจำนวนรุ่นที่จำกัดไว้	591.6
shift 0.7	2.58	ถึงจำนวนรุ่นที่จำกัดไว้	580.8
shift-and-check 0.7	2.61	ถึงจำนวนรุ่นที่จำกัดไว้	597.6
shift 0.8	2.58	ถึงจำนวนรุ่นที่จำกัดไว้	506.0
shift-and-check 0.8	2.65	ถึงจำนวนรุ่นที่จำกัดไว้	546.8
shift 0.9	2.53	ถึงจำนวนรุ่นที่จำกัดไว้	577.6
shift-and-check 0.9	2.67	ถึงจำนวนรุ่นที่จำกัดไว้	618.4
shift 1.0	2.52	ถึงจำนวนรุ่นที่จำกัดไว้	539.2
shift-and-check 1.0	2.67	ถึงจำนวนรุ่นที่จำกัดไว้	575.6

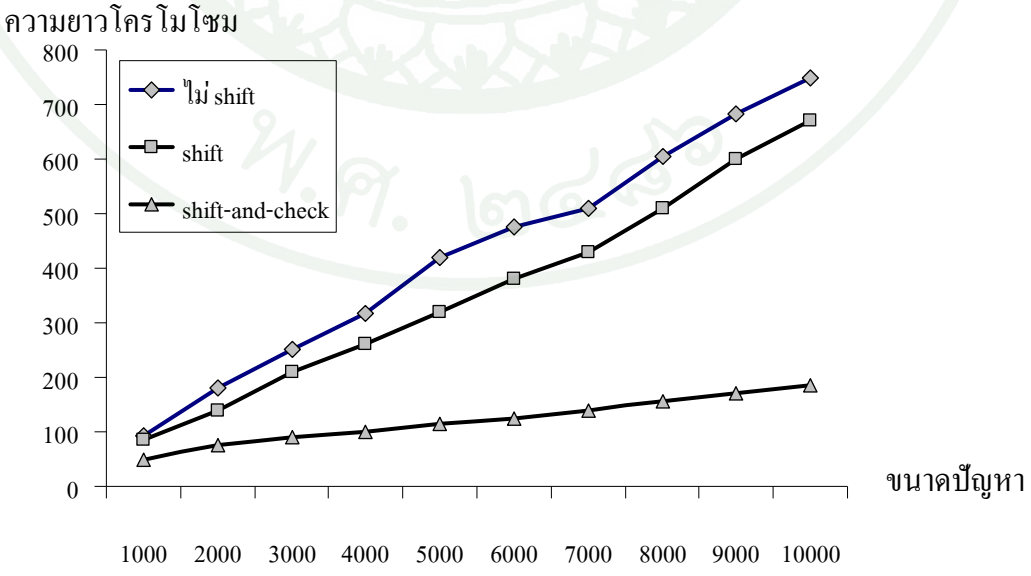
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

1.2 ผลจากการทดลองเพื่อเปรียบเทียบความยาวโครโมโซมที่ใช้ในการค้นหาคำตอบ

ผลจากการทดลองเพื่อเปรียบเทียบความยาวโครโมโซมที่ใช้ในการค้นหาคำตอบโดยใช้ปัญหา OneMax แสดงในตารางที่ 25 จะเห็นว่าตัวดำเนินการ shift และ shift-and-check ช่วยให้ LZWGA ใช้ความยาวโครโมโซมลดลงเฉลี่ย 16.28% และ 68.05% ตามลำดับ

ตารางที่ 25 ความยาวของโครโมโซม LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา OneMax

ขนาดปัญหา	ไม่ shift	shift		shift-and-check	
		ความยาวที่ใช้	ลดลง(%)	ความยาวที่ใช้	ลดลง(%)
1000	92	85	7.61	50	45.65
2000	180	140	22.22	75	58.33
3000	252	210	16.67	90	64.29
4000	318	260	18.24	100	68.55
5000	420	320	23.81	115	72.62
6000	476	380	20.17	125	73.74
7000	510	430	15.69	140	72.55
8000	605	510	15.70	155	74.38
9000	682	600	12.02	170	75.07
10000	750	670	10.67	185	75.33



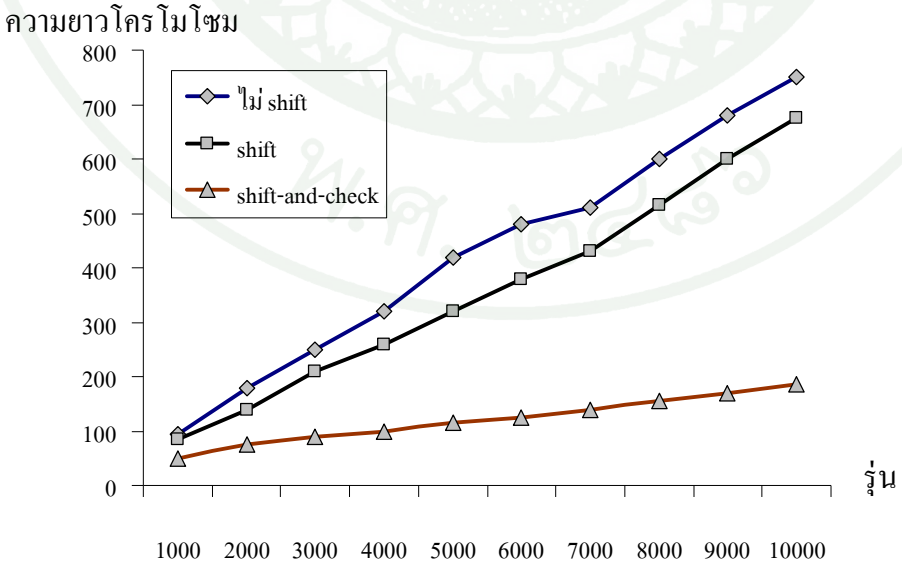
ภาพที่ 23 ความยาวโครโมโซมของ LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา OneMax

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ผลจากการทดลองเพื่อเปรียบเทียบความยาวโครโมโซมที่ใช้ในการค้นหาคำตอบโดยใช้ปัญหา RoyalRoad แสดงในตารางที่ 26 จะเห็นว่าตัวดำเนินการ shift และ shift-and-check ช่วยให้ LZWGA ใช้ความยาวโครโมโซมลดลงเฉลี่ย 16.38% และ 68.21% ตามลำดับ

ตารางที่ 26 ความยาวของโครโมโซม LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา RoyalRoad

ขนาดปัญหา	ไม่ shift	shift		shift-and-check	
		ความยาวที่ใช้	ลดลง(%)	ความยาวที่ใช้	ลดลง(%)
1000	95	85	10.53	50	47.37
2000	180	140	22.22	75	58.33
3000	250	210	16.00	90	64.00
4000	320	260	18.75	100	68.75
5000	420	320	23.81	115	72.62
6000	480	380	20.83	125	73.96
7000	510	430	15.69	140	72.55
8000	600	515	14.17	155	74.17
9000	680	600	11.76	170	75.00
10000	750	675	10.00	185	75.33



ภาพที่ 24 ความยาวโครโมโซมของ LZWGA ที่ใช้ในการค้นหาคำตอบปัญหา RoyalRoad

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

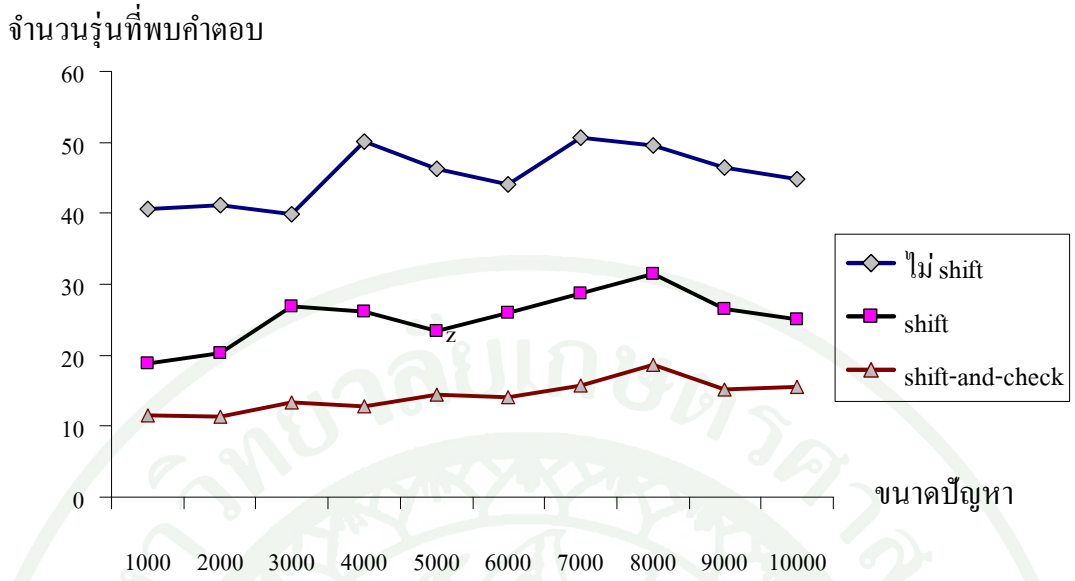
1.3 ผลจากการทดลองเพื่อเปรียบเทียบจำนวนรุ่นที่ใช้ในการพบคำตอบ

ผลจากการทดลองเพื่อเปรียบเทียบจำนวนรุ่นที่พบคำตอบโดยใช้ปัญหา OneMax แสดงในตารางที่ 27 จะเห็นว่าตัวดำเนินการ shift และ shift-and-check ช่วยให้ LZWGA ใช้จำนวนรุ่นในการค้นหาคำตอบลดลงเฉลี่ย 44.19% และ 68.63% ตามลำดับ

ตารางที่ 27 จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา OneMax

ขนาดปัญหา	ไม่ shift	shift		shift-and-check	
		จำนวนรุ่นที่ใช้	ลดลง(%)	จำนวนรุ่นที่ใช้	ลดลง(%)
1000	40.7	18.9	53.56	11.5	71.74
2000	41.2	20.3	50.73	11.3	72.57
3000	39.8	26.8	32.66	13.3	66.58
4000	50.1	26.2	47.70	12.8	74.45
5000	46.2	23.4	49.35	14.4	68.83
6000	44	25.9	41.14	14.1	67.95
7000	50.7	28.7	43.39	15.7	69.03
8000	49.5	31.4	36.57	18.6	62.42
9000	46.4	26.6	42.67	15.2	67.24
10000	44.9	25.1	44.10	15.5	65.48

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



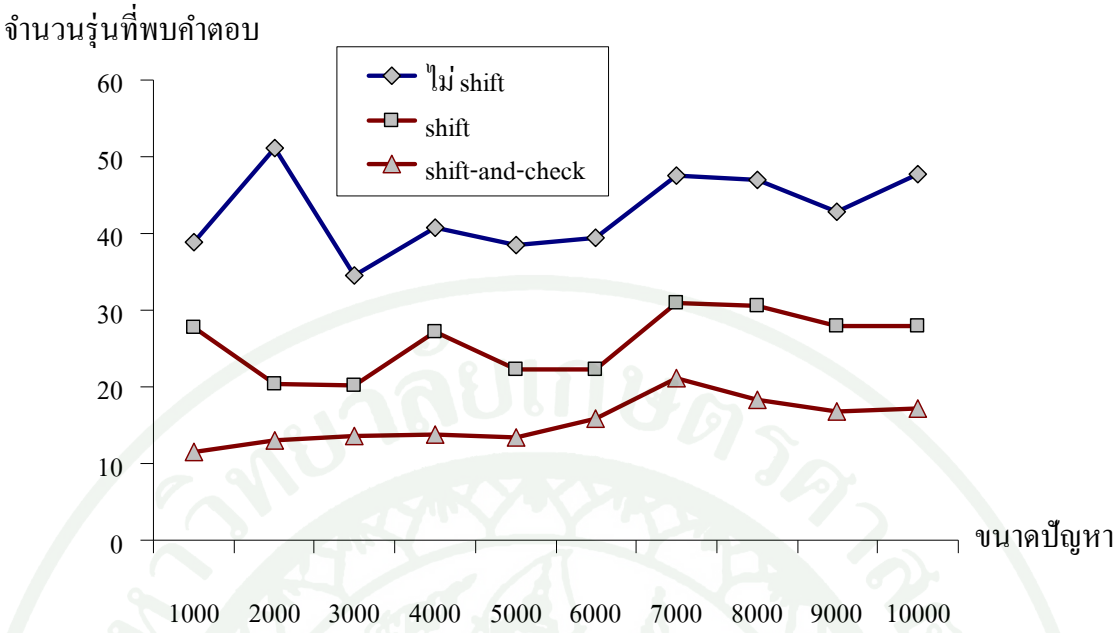
ภาพที่ 25 จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา OneMax

ผลจากการทดลองเพื่อเปรียบเทียบจำนวนรุ่นที่โครโมโซม LZWGA พบคำตอบในปัญหา RoyalRoad แสดงในตารางที่ 28 จะเห็นว่าตัวดำเนินการ shift และ shift-and-check ช่วยให้ LZWGA ใช้จำนวนรุ่นในการค้นหาคำตอบลดลงเฉลี่ย 39.60% และ 63.83% ตามลำดับ

ตารางที่ 28 จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา RoyalRoad

ขนาดปัญหา	ไม่ shift	shift		shift-and-check	
		จำนวนรุ่นที่ใช้	ลดลง(%)	จำนวนรุ่นที่ใช้	ลดลง(%)
1000	38.8	27.7	28.61	11.6	70.10
2000	51.2	20.3	60.35	13.1	74.41
3000	34.5	20.2	41.45	13.6	60.58
4000	40.8	27.1	33.58	13.7	66.42
5000	38.5	22.3	42.08	13.4	65.19
6000	39.5	22.3	43.54	15.8	60.00
7000	47.5	30.9	34.95	21.1	55.58
8000	46.9	30.5	34.97	18.3	60.98
9000	42.9	27.9	34.97	16.8	60.84
10000	47.7	27.9	41.51	17.1	64.15

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 26 จำนวนรุ่นที่โครโมโซม LZWGA ที่ใช้ในการพบคำตอบปัญหา RoyalRoad

2. ผลการทดลองเพื่อวัดประสิทธิภาพของการปรับปรุงตัวดำเนินการ shift-and-check (ISC)

ผลการทดลองกับปัญหา OneMax และ RoyalRoad พบว่าทั้งตัวดำเนินการ shift-and-check และ ISC ต่างก็ช่วยให้ LZWGA พบคำตอบได้ ตารางที่ 29 แสดงจำนวนรุ่นเฉลี่ยที่พบคำตอบ โดยตัวดำเนินการ ISC ช่วยให้ LZWGA พบคำตอบได้เร็วกว่าตัวดำเนินการ shift-and-check 1.40 และ 1.33 เท่าในปัญหา OneMax และ RoyalRoad ตามลำดับ

ตารางที่ 29 จำนวนรุ่นที่พบคำตอบ

ปัญหา	OneMax	RoyalRoad
LZWGA ธรรมดา	ไม่พบ	ไม่พบ
LZWGA ที่ใช้ตัวดำเนินการ shift-and-check	951	998
LZWGA ที่ใช้ตัวดำเนินการ ISC	682	751

สำหรับปัญหา RandomRoyalRoad LZWGA หาคำตอบไม่พบ ค่าความเหมาะสมของโครโมโซมที่ดีที่สุด แสดงในตารางที่ 30 จากตารางจะเห็นว่า LZWGA ที่ใช้ตัวดำเนินการ ISC

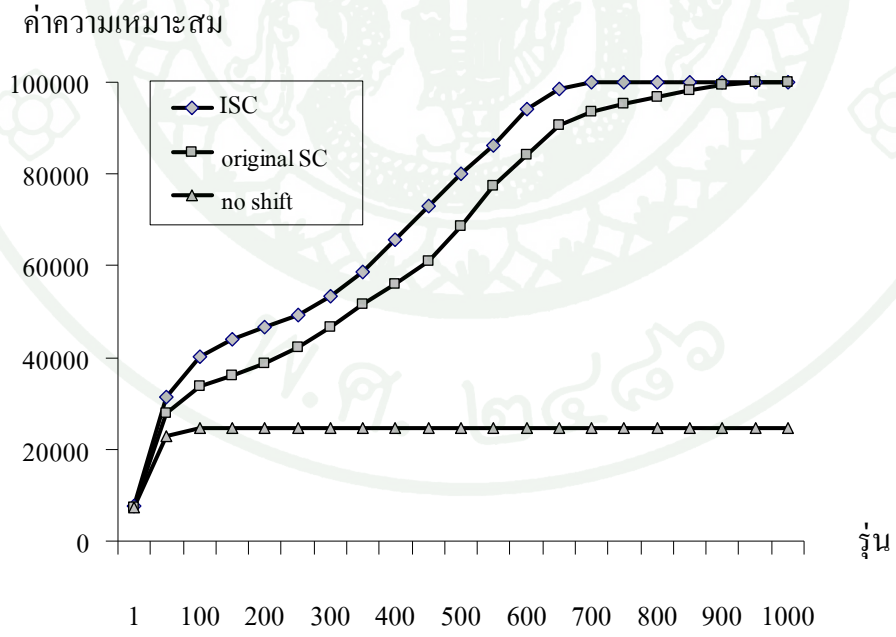
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สร้างโครโมโซมที่มีค่าความเหมาะสมมากกว่า shift-and-check 1.09 เท่า และสร้างโครโมโซมที่มีค่าความเหมาะสมมากกว่า LZWGA ธรรมดา 1.70 เท่า

ตารางที่ 30 ค่าความเหมาะสมของโครโมโซมที่ดีที่สุดในปัญหา RandomRoyalRoad

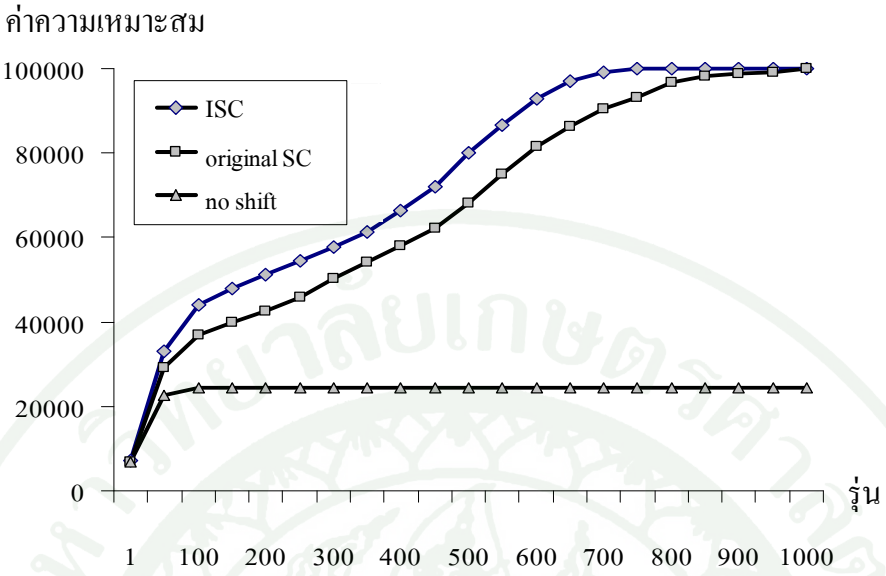
ปัญหา	ค่าความเหมาะสม
LZWGA ธรรมดา	6,256
LZWGA ที่ใช้ตัวดำเนินการ shift-and-check	9,778
LZWGA ที่ใช้ตัวดำเนินการ ISC	10,616

ค่าความเหมาะสมของโครโมโซมที่ดีที่สุดในแต่ละรุ่นของวิวัฒนาการในทั้ง 3 ปัญหา แสดงดังภาพที่ 27, 28 และ 29 ตามลำดับ จะเห็นว่า LZWGA ธรรมดาสร้างโครโมโซมที่มีค่าความเหมาะสมต่ำที่สุด ส่วน LZWGA ที่ใช้ตัวดำเนินการ ISC นั้น จะสร้างโครโมโซมที่มีค่าความเหมาะสมสูงกว่า LZWGA ที่ใช้ตัวดำเนินการ shift-and-check

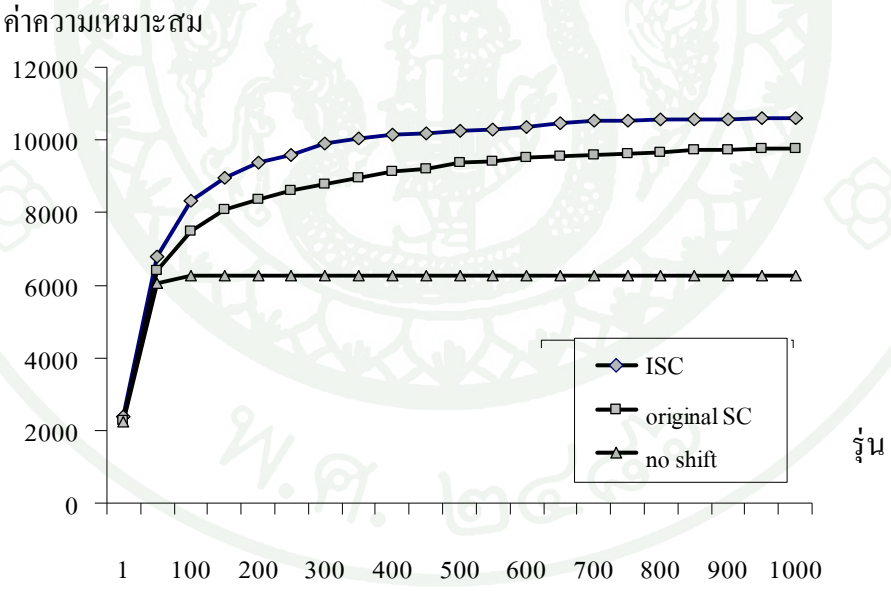


ภาพที่ 27 ค่าความเหมาะสมของโครโมโซมของ LZWGA ที่ได้ในปัญหา OneMax

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 28 ค่าความเหมาะสมของโครโมโซมของ LZWGA ที่ได้ในปัญหา RoyalRoad



ภาพที่ 29 ค่าความเหมาะสมของโครโมโซมของ LZWGA ที่ได้ในปัญหา RandomRoad

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3. ผลการทดลองเพื่อวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการ Shift และ Shift-and-check ใน LZWGA

ผู้วิจัยได้วิเคราะห์ประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check (อณิมา และวรเศรษฐ, 2552) โดยที่ผลการวิเคราะห์ประสิทธิภาพพบว่าตัวดำเนินการ shift และ shift-and-check ทำให้โครโมโซมส่วนใหญ่มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม ดังแสดงในตารางที่ 31 และ 32 (*n* คือ ความยาวโครโมโซม, *TC* คือ จำนวนโครโมโซมทุกรูปแบบที่เป็นไปได้และนำมาผ่านการ shift ทุกตำแหน่ง) เมื่อกำหนดโครโมโซม LZWGA ขนาด 4 ตำแหน่ง พบว่าถ้านำโครโมโซมมา shift-and-check ทุกตำแหน่ง จะทำให้เกิดโครโมโซมที่เป็นไปได้ทั้งหมด 480 แบบ (4×(4+1)!) โดยจะมีโครโมโซมที่มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมถึง 441 แบบในปัญหา OneMax, 363 แบบในปัญหา RandomMax, 475 แบบในปัญหา RoyalRoad, และ 471 แบบในปัญหา RandomRoyalRoad

ตารางที่ 31 จำนวนโครโมโซมที่ shift แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมในปัญหาทดสอบ

n	***TC***	**จำนวนโครโมโซมที่ Shift แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม**			
		OneMax	**RandomMax**	**RoyalRoad**	**Random RoyalRoad**
4	480	287.0	357.0	411.0	471.0
5	3,600	1,921.0	2,545.0	2,918.0	3,484.0
6	30,240	14,282.0	21,106.0	23,284.0	28,567.0
7	282,240	123,754.0	189,484.0	211,649.0	262,522.0
8	2,903,040	1,153,254.0	1,881,839.0	2,127,639.0	2,642,015.0

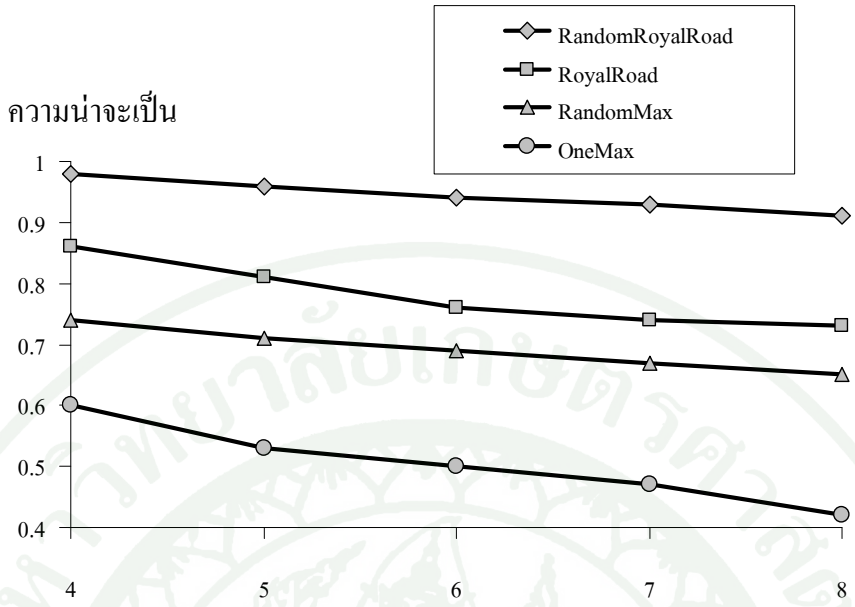
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตารางที่ 32 จำนวนโครโมโซมที่ shift-and-check แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมในปัญหาทดสอบ

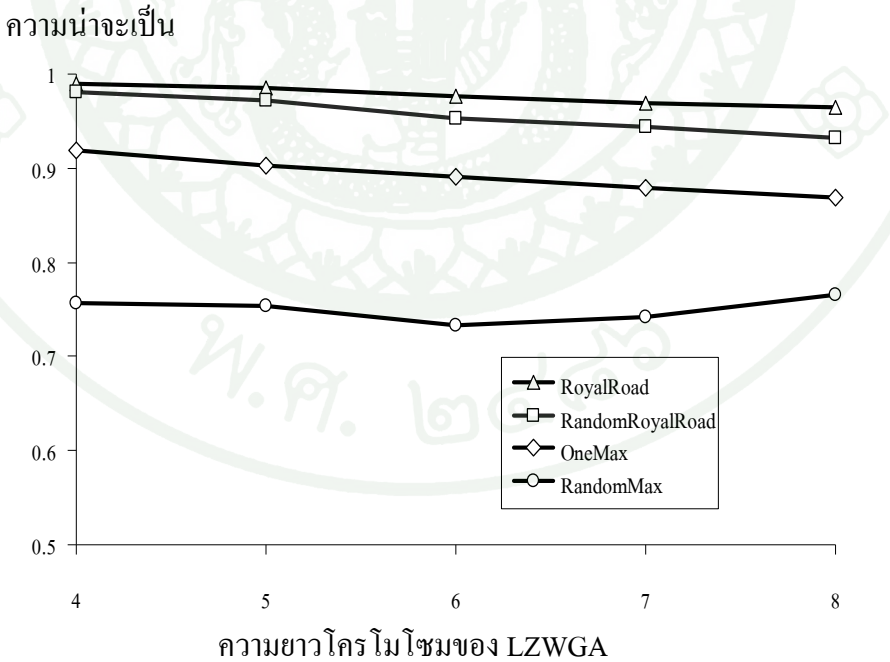
n	***TC***	**จำนวนโครโมโซมที่ Shift-and-Check แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม**			
		OneMax	**RandomMax**	**RoyalRoad**	**Random RoyalRoad**
4	480	441	363	475	471
5	3,600	3,252	2,715	3,548	3,497
6	30,240	26,928	22,174	29,540	28,825
7	282,240	248,157	209,327	273,577	266,326
8	2,903,040	2,523,477	2,220,282	2,802,036	2,704,113

เมื่อนำข้อมูลจากตารางที่ 31 และ 32 มาคำนวณค่าความน่าจะเป็นแล้ววาดกราฟจะได้ดังภาพที่ 30 และ 31 ซึ่งแสดงค่าความน่าจะเป็นที่ตัวดำเนินการ shift-and-check สร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้นหรือเท่าเดิมในปัญหาต่างๆ ที่ความยาวโครโมโซมตั้งแต่ 4 ถึง 8 ตำแหน่ง เมื่อโครโมโซม LZWGA มีความยาวมากขึ้น ความน่าจะเป็นที่การ shift และ shift-and-check จะทำให้โครโมโซมมีค่าความเหมาะสมเพิ่มขึ้นจะค่อยๆ ลดลง

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



ภาพที่ 30 ความน่าจะเป็นที่การ shift ทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้นหรือเท่าเดิม



ภาพที่ 31 ความน่าจะเป็นที่การ shift-and-check ทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้นหรือเท่าเดิม

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ในการทดสอบกับปัญหา OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad พบว่า ตัวดำเนินการ shift สามารถสร้างโครโมโซมที่ดีขึ้นหรือเท่ากับโครโมโซมเดิมด้วยความน่าจะเป็นเฉลี่ยเท่ากับ 0.50, 0.69, 0.78, และ 0.94 ตามลำดับ ส่วนตัวดำเนินการ shift-and-check สามารถสร้างโครโมโซมที่ดีขึ้นหรือเท่ากับโครโมโซมเดิมด้วยความน่าจะเป็นเฉลี่ยเท่ากับ 0.89, 0.75, 0.97, และ 0.95 ตามลำดับ เนื่องจากความน่าจะเป็นดังกล่าวมีมากกว่า 0.5 จึงส่งผลให้ LZWGA พบคำตอบเร็วขึ้นหรือได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นเมื่อใช้ตัวดำเนินการ shift-and-check

วิจารณ์

การวิจารณ์ผลการทดลองแบ่งออกเป็น 3 กรณีคือ วิจารณ์ผลการวัดประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check, วิจารณ์ผลการวัดประสิทธิภาพของการปรับปรุงตัวดำเนินการ shift-and-check และวิจารณ์ผลการวิเคราะห์ประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check

1. วิจารณ์ผลการวัดประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check

ผลการทดลองวัดประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check เป็นไปตามสมมติฐานคือ ตัวดำเนินการ shift และ shift-and-check ต่างก็ช่วยให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้นในทุกรุ่นของวิวัฒนาการ ซึ่งในปัญหา OneMax และ RoyalRoad นั้น LZWGA ที่ใช้ตัวดำเนินการ shift-and-check สามารถวิวัฒนาการจนพบคำตอบได้ ส่วน LZWGA ที่ใช้ตัวดำเนินการ shift ไม่สามารถหาคำตอบได้พบเนื่องมาจากการทดลองนี้กระทำกับกลุ่มประชากรขนาดเล็ก ผู้วิจัยได้เคยทดลองใช้กลุ่มประชากรขนาดใหญ่ (1,000) ซึ่งตัวดำเนินการ shift ก็สามารถช่วยให้ LZWGA หาคำตอบพบได้เช่นกันถึงแม้จะไม่ใช่ทุกครั้งก็ตาม

นอกจากตัวดำเนินการ shift และ shift-and-check จะช่วยให้โครโมโซมมีค่าความเหมาะสมเพิ่มขึ้นแล้ว ยังช่วยให้ LZWGA หาคำตอบได้โดยใช้ความยาวโครโมโซมสั้นกว่า LZWGA และใช้จำนวนรุ่นในการค้นหาคำตอบน้อยกว่ามาก นั่นคือ ถ้าเปรียบเทียบเวลาที่ใช้ในการค้นหาคำตอบถึงแม้ตัวดำเนินการ shift และ shift-and-check จะมีขั้นตอนในการทำงานเพิ่มขึ้น แต่ก็ช่วยทำให้ LZWGA ค้นพบคำตอบได้เร็วขึ้นและสิ้นสุดการทำงานได้เร็วขึ้นด้วย

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ปัญหา RandomMax และ RandomRoyalRoad เป็นปัญหาที่ยากสำหรับ LZWGA เพราะสร้างโครโมโซมที่มีค่าความเหมาะสมต่ำ ตัวดำเนินการ shift และ shift-and-check ต่างก็ทำให้ LZWGA สร้างโครโมโซมที่มีค่าความเหมาะสมเพิ่มขึ้น แต่ทั้งนี้ก็ใช้เวลาในการประมวลผลเพิ่มขึ้นด้วยเช่นกัน

การทดลองพิจารณาอัตราการ shift ที่ 0.1 ถึง 1.0 ผลการทดลองแสดงให้เห็นว่าอัตราการ shift มากมิได้ทำให้โครโมโซมที่ได้มีค่าความเหมาะสมสูงเสมอไป เนื่องจากเมื่ออัตราการ shift มากโอกาสที่โครโมโซมจะคลาดเคลื่อนจากตำแหน่งที่ถูกต้องก็มีมากตามไปด้วย ดังนั้นอัตราการ shift ที่เหมาะสมจึงควรอยู่ในระดับกลางคือ ประมาณ 0.5

2. วิจารณ์ผลการวัดประสิทธิภาพของการปรับปรุงตัวดำเนินการ shift-and-check

การวัดประสิทธิภาพของการปรับปรุงตัวดำเนินการ shift-and-check (ISC) มุ่งเน้นในการแก้ปัญหาขนาดใหญ่โดยใช้กลุ่มประชากรขนาดเล็ก ผลการทดลองพบว่า LZWGA ที่ใช้ตัวดำเนินการ ISC สามารถสร้างโครโมโซมที่มีค่าความเหมาะสมสูงกว่า LZWGA ที่ใช้ ตัวดำเนินการ shift-and-check อีกทั้งสามารถพบคำตอบได้เร็วกว่าอีกด้วย

ทั้งตัวดำเนินการ ISC และ shift-and-check ต่างก็ช่วยให้ LZWGA หาคำตอบพบ แต่ตัวดำเนินการ ISC ช่วยให้พบคำตอบได้เร็วกว่า ดังนั้นถ้าหากในการทำงานกับปัญหาขนาดใหญ่ที่มีกลุ่มประชากรขนาดเล็ก และจำกัดเวลาในการทดลองรวมถึงจำนวนรุ่นที่ใช้ในการค้นหาคำตอบ ตัวดำเนินการ ISC จะช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงกว่า และช่วยเพิ่มประสิทธิภาพการทำงานให้ LZWGA ได้มากกว่าตัวดำเนินการ shift-and-check

3. วิจารณ์ผลการวิเคราะห์ประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check

จากการทดลองเพื่อวิเคราะห์ประสิทธิภาพของตัวดำเนินการ shift และ shift-and-check เราพบว่าทั้งตัวดำเนินการ shift และ shift-and-check ต่างก็มีความน่าจะเป็นเฉลี่ยมากกว่า 0.5 ที่จะสร้างโครโมโซมที่มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม ในปัญหา OneMax ซึ่งเป็นปัญหาที่ง่ายที่สุด เราพบว่าความน่าจะเป็นน้อยที่สุดที่ตัวดำเนินการ shift จะสร้างโครโมโซมที่มีค่าความเหมาะสมดีขึ้นหรือเท่าเดิม สาเหตุมาจากการที่ในปัญหา OneMax นี้โครโมโซมที่สร้างขึ้นมามีค่าความ

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

เหมาะสมดีกว่าโครโมโซมในปัญหาอื่นๆมากอยู่แล้ว ดังนั้นโอกาสที่ค่าความเหมาะสมของมันจะเพิ่มขึ้นจึงน้อยกว่าปัญหาอื่นๆ ในขณะที่ปัญหายากๆ เช่น RandomRoyalRoad แต่เดิมที่สร้างโครโมโซมขึ้นค่าความเหมาะสมของมันมีน้อย ดังนั้นโอกาสที่การ shift จะทำให้มีค่าความเหมาะสมเพิ่มขึ้นจึงมีมากกกว่าเช่นกัน

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

สรุปและข้อเสนอแนะ

สรุป

วัตถุประสงค์ของงานวิจัยนี้คือเพื่อพัฒนาตัวดำเนินการ shift ที่จะช่วยปรับปรุงประสิทธิภาพการทำงานของ LZWGA ให้สามารถหาคำตอบพบได้โดยใช้ความยาวโครโมโซมของ LZWGA ที่สั้นลง และใช้จำนวนรุ่นในการพบคำตอบน้อยลง ส่วนในปัญหาที่หาคำตอบไม่พบจะต้องสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น รวมถึงเมื่อได้ตัวดำเนินการดังกล่าวแล้ว เราจะต้องวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการนี้ใน LZWGA ด้วย

ตัวดำเนินการที่ถูกพัฒนาคือตัวดำเนินการ shift และ shift-and-check ซึ่งต่างก็ช่วยเพิ่มประสิทธิภาพให้กับ LZWGA ในทุกปัญหาที่นำมาทดสอบ ปัญหาที่สามารถหาคำตอบพบคือ OneMax และ RoyalRoad ส่วนปัญหาที่หาคำตอบไม่พบคือ RandomMax และ RandomRoyalRoad แต่ก็สามารถสร้างโครโมโซมที่มีค่าความเหมาะสมเพิ่มขึ้น สาเหตุที่ตัวดำเนินการ shift และ shift-and-check ช่วยให้ LZWGA ทำงานได้อย่างมีประสิทธิภาพ นั่นคือความน่าจะเป็นที่การ shift และ shift-and-check จะช่วยทำให้โครโมโซมที่ได้มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมมีค่ามากกว่า 0.5

การปรับปรุงตัวดำเนินการ shift-and-check (ISC) ให้เหมาะกับในปัญหาขนาดใหญ่นั้นก็เช่นเดียวกัน คือช่วยให้ LZWGA สามารถหาพบคำตอบได้เร็วขึ้นโดยใช้จำนวนรุ่นน้อยลง ส่วนในปัญหาที่หาคำตอบไม่พบก็ช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงกว่า LZWGA ที่ใช้ตัวดำเนินการ shift-and-check

ปัญหาที่เกิดจากการทดลองนี้คือ ตัวดำเนินการ shift สร้างโครโมโซมที่มีค่าความเหมาะสมไม่สูงมาก นั่นเป็นเพราะการทดลองกระทำกับกลุ่มประชากรขนาดเล็ก ผู้วิจัยได้เคยทดลองใช้ตัวดำเนินการ shift กับปัญหาที่มีกลุ่มประชากรขนาดใหญ่มากขึ้น ซึ่งค่าความเหมาะสมที่ได้จากตัวดำเนินการ shift จัดว่าอยู่ในเกณฑ์สูง แต่เมื่อประชากรมีขนาดใหญ่ก็จะทำให้สิ้นเปลืองเวลาประมวลผลตามไปด้วย กล่าวโดยสรุป จากผลการทดลองตัวดำเนินการ shift-and-check ทำงานได้ดีในปัญหาทุกขนาดที่มีกลุ่มประชากรขนาดเล็ก

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ข้อเสนอแนะ

งานวิจัยนี้มุ่งเน้นการหาคำตอบโดยใช้ LZWGA ซึ่งเป็นอัลกอริทึมที่พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม ซึ่งในปัญหาที่ง่าย เช่น ปัญหา OneMax ตัวดำเนินการของเราช่วยสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้น และช่วยทำให้พบคำตอบได้เร็วขึ้น แต่ในปัญหาที่ยากเช่น RandomRoyalRoad เป็นที่น่าศึกษาหาเหตุผลที่ LZWGA ถึงหาคำตอบไม่พบและควรศึกษาหาวิธีการแก้ไขปัญหานี้ต่อไป

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

เอกสารและสิ่งอ้างอิง

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

วรเศรษฐ สุวรรณิก, นริศ กุณาศล และ ประภาส จงสถิตย์วัฒนา. 2548. ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด, น. 203-211. ใน **Proceedings of National Computer Science and Engineering Conference (NCSEC 2005)**. มหาวิทยาลัยหอการค้าไทย, กรุงเทพฯ.

สมรักษ์ นุ่มนาค และ วรเศรษฐ สุวรรณิก. 2550. การปรับปรุงการกลายพันธุ์ใน LZWGA: กรณีศึกษาการลดความยาวโครโมโซมที่ถูกบีบอัดในปัญหา OneMax, น. 612-615. ใน **ประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 30**. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี, กรุงเทพฯ.

_____ และ _____. 2550. เปรียบเทียบประสิทธิภาพวิธีการไขว้เปลี่ยนใน LZWGA, น. 328-333. ใน **การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 3**. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, กรุงเทพฯ.

สาวิตรี ฉ่ำลมัย และ วรเศรษฐ สุวรรณิก. 2550. ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด, น. 653-656. ใน **ประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 30**. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี, กรุงเทพฯ.

อณิมา รอดเสียงล้ำ และ วรเศรษฐ สุวรรณิก. 2552. การวิเคราะห์ประสิทธิภาพของตัวดำเนินการ Shift-and-Check ใน LZWGA. ใน **ประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 32 (EECON32)**. มหาวิทยาลัยมหิดล, กรุงเทพฯ.

_____ และ _____ . 2552. ตัวดำเนินการ Shift สำหรับ LZWGA, น. 320-325. ใน **การประชุมทางวิชาการระดับชาติทางด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 5**. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, กรุงเทพฯ.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

_____ และ _____ . 2552. การเปรียบเทียบประสิทธิภาพการทำงานของ LZWGA บนวิธีการคัดเลือกที่แตกต่าง, น. 198-203. ใน **Proceedings of National Computer Science and Engineering Conference (NCSEC 2009)**. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี, กรุงเทพฯ.

Goldberg, D.E. 1989. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Addison-Wesley Professional, USA.

Jadaan, O.A., L. Rajamani and C.R. Rao. 2005. Improved Selection Operator for GA. pp. 269-277. In **Journal of Theoretical Information Technology**.

Feng, L. and Y. Qi-wen. 2002. Improved Genetic Operator for Genetic Algorithm, pp. 431-434. In **Journal of Zhejiang University SCIENCE**.

Kunasol, N., W. Suwannik and P. Chongstitvatana. 2006. Solving One-Million-Bit Problems Using LZWGA. pp. 32-36. In **Proceedings of International Symposium on Communications and Information Technologies (ISCIT 2006)**. King Mongkut's Institute of Technology Ladkrabang, Thailand.

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

Numnark, S. and W. Suwannik. 2008. Improving the Performance of LZWGA by Using a New Mutation Method, pp. 1862-1865. In **Proceedings of IEEE Congress on Evolutionary Computation (CEC 2008)**. The Chinese University of Hong Kong, Hong Kong.

Rasheed, K., 1999. Guided Crossover: A New Operator for Genetic Algorithm Based Optimization, pp. 1535-1541. In **Proceeding of the Congress on Evolutionary Computation**.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

Rotsianglum, A. and W.Suwannik. 2010. An Improved Shift-and-Check Operator for LZWGA. In **The 7th International Joint conference on Computer Science and Software Engineering (JCSSE).**

Suwannik, W. and P.Chongstitvatana. 2007. Solving Large Scale Problems using Estimation Distribution Algorithm with Arithmetic Coding, pp. 358-363. In **Proceedings of International Symposium on Communications and Information Technologies (ISCIT)**.

Wang, R.L. and K. Okazaki. 2007. An Improved Genetic Algorithm with Conditional Genetic Operators and its Application to Set-Covering Problem, pp. 687-694. In **Soft Comput**.

Yokoo, H. 1992. Improved Variations Relating the Ziv-Lempel and Welch-Type Algorithm for Sequential Data Compression, pp. 73-81. In **IEEE Trans on Information Theory**.

Zhong, J., X. Hu, M. Gu and J. Zhang. 2005. Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms. In **Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC)**.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ภาคผนวก

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ตัวดำเนินการ Shift สำหรับ LZWGA
A Shift Operator for LZWGA

อณิมา รอดเสียงล้ำ และ วรเศรษฐ สุวรรณิก

ภาควิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
g5164149@ku.ac.th, worasait.suwannik@gmail.com

บทคัดย่อ

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) โดยเข้ารหัสโครโมโซมในรูปของการบีบอัด LZWGA ช่วยลดขนาดของโครโมโซมและสามารถแก้ปัญหาขนาดใหญ่มากได้ แต่เมื่อจำกัดความยาวโครโมโซมของ LZWGA ให้กลายได้พอดีกับจำนวนบิตของคำตอบใน GA แล้วทำให้ไม่สามารถพบคำตอบได้ บทความนี้เสนอตัวดำเนินการทางพันธุกรรมตัวใหม่เรียกว่า "shift" ซึ่งเป็นการเลื่อนตำแหน่งของโครโมโซม LZWGA หลังตำแหน่งที่สุ่มได้ทั้งหมดมาทางซ้ายหนึ่งตำแหน่ง ผลการทดลองกับปัญหาที่จำกัดความยาวโครโมโซมพบว่าตัวดำเนินการ shift ช่วยให้ LZWGA สามารถพบคำตอบได้และสำหรับปัญหาที่ LZWGA หาคำตอบไม่พบ ตัวดำเนินการ shift จะช่วยให้โครโมโซมที่ได้มีค่าความเหมาะสมสูงขึ้น

คำสำคัญ: การเข้ารหัสแบบ LZW ขั้นตอนวิธีเชิงพันธุกรรม ตัวดำเนินการ

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. This encoding reduces the length of the chromosome and enables the algorithm to solve very large problems. This paper presents a new operator called "shift". A shift operator randomly selects a shift position and left-shifts numbers in an LZWGA chromosome. The result shows that a shift operator for LZWGA can solve problems with a shorter chromosome length. For problems, which original LZWGA can not find a result, a shift operator will produce a chromosome that has better fitness value.

Keywords: LZWGA, GA, Operator

1. บทนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm - GA) เป็นวิธีการค้นหาที่เลียนแบบวิวัฒนาการของสิ่งมีชีวิต [1] การค้นหากระทำกับกลุ่มของโครโมโซมที่ถูกเข้ารหัสเป็นเลขฐานสอง (binary string) เมื่อโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหาก็จะมีขนาดใหญ่ขึ้นตามไปด้วย เช่นปัญหา OneMax ขนาด 128 บิต มีคำตอบที่เป็นไปได้ 2^{128} แบบ ขณะที่ปัญหา OneMax หนึ่งล้านบิต จะมีคำตอบที่เป็นไปได้ทั้งหมด $2^{1000000}$ แบบ

LZWGA เข้ารหัสโครโมโซมแบบบีบอัดทำให้โครโมโซมมีขนาดเล็กลง การวิวัฒนาการทำกับข้อมูลที่ถูกบีบอัดซึ่งเป็นอาร์เรย์ของเลขจำนวนเต็ม เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำมาคลายจะได้โครโมโซมของ GA คือ 1010101 ซึ่งค่าความเหมาะสม (fitness) จะวัดกับโครโมโซมที่คลายแล้ว [2]

LZWGA สามารถแก้ปัญหาที่มีขนาดใหญ่มากถึงหนึ่งล้านบิต [3] ปัญหาดังกล่าวคือ OneMax, RoyalRoad, และ Trap

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

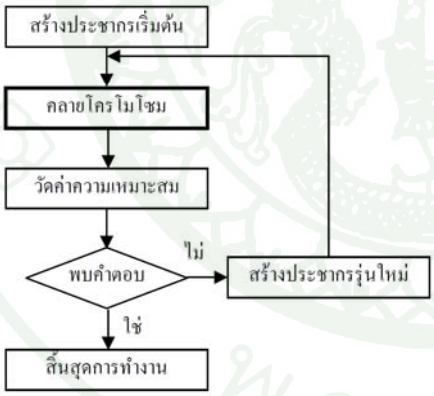
ซึ่งคำตอบของปัญหาดังกล่าวมีแบบแผนสูง (คำตอบเป็น 1 หมดทุกบิต) แต่สำหรับปัญหาที่ไม่มีแบบแผน เช่น ปัญหา RandomMax และ RandomRoyalRoad ที่มีค่าความสุ่มเป็น 1 LZWGA ไม่สามารถหาคำตอบพบ นอกจากนั้นยังได้ค่าความเหมาะสมของโครโมโซมเฉลี่ยน้อยกว่า GA [4]

โครโมโซมของ LZWGA เป็นอาร์เรย์ของเลขจำนวนเต็ม การเลื่อนโครโมโซมเพียงหนึ่งตำแหน่งอาจทำให้โครโมโซมมีค่าความเหมาะสมที่สูงขึ้นหรือลดลงได้ดังแสดงในส่วนที่ 4 บทความนี้จึงเสนอการใช้ตัวดำเนินการ shift ซึ่งเป็นการเลื่อนตำแหน่งของโครโมโซมหลังตำแหน่งที่สุ่มได้ทั้งหมดมาทางซ้ายหนึ่งตำแหน่ง โดยเราทดลองตัวดำเนินการนี้กับปัญหาที่มีแบบแผนและที่ไม่มีแบบแผน

2. LZWGA

2.1 ขั้นตอนการทำงาน

การทำงานของ LZWGA คล้ายกับการทำงานของ GA โดย LZWGA เพิ่มขั้นตอนการคลายโครโมโซมก่อนที่จะนำมาวัดค่าความเหมาะสม ดังแสดงในภาพที่ 1



ภาพที่ 1 : ขั้นตอนการทำงานของ LZWGA

2.2 ประชากร

โครโมโซมของ GA จะเป็นเลขฐานสอง ส่วนโครโมโซมที่ถูกบีบอัดของ LZWGA จะเป็นอาร์เรย์ของเลขจำนวนเต็มประกอบด้วยเลขตั้งแต่ 0-n โดย n คือขนาดของโครโมโซมที่ถูกบีบอัด การสร้างโครโมโซมรุ่นแรกของ LZWGA จะต้องสุ่มให้อยู่ในเงื่อนไขที่สามารถนำไปคลายได้ เงื่อนไขดังกล่าวคือ ตำแหน่งที่ 1 ของโครโมโซมที่ถูกบีบอัดจะมีค่าเป็น 0 หรือ 1 ส่วนตำแหน่งถัดไปจะมีค่าได้ไม่เกินตำแหน่งนั้น ตัวอย่างเช่น โครโมโซมตำแหน่งที่ 1 ค่าที่เป็นไปได้คือ {0, 1} โครโมโซมตำแหน่งที่ 2 ค่าที่เป็นไปได้คือ {0, 1, 2} ซึ่งสามารถสรุปได้ดังสมการที่ 1

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

โดย a_i คือ ตำแหน่งที่ i ของโครโมโซมที่ถูกบีบอัดอยู่

r คือ เลขที่สุ่มขึ้นมา

n คือ ขนาดของโครโมโซม

2.3 การคลายและการวัดค่าความเหมาะสม

โครโมโซมของ LZWGA อยู่ในรูปของการบีบอัด ดังนั้นก่อนวัดค่าความเหมาะสม จึงต้องมีการคลายโครโมโซมก่อน ตัวอย่างเช่น โครโมโซมของ LZWGA คือ {0, 2, 1, 3, 1, 1} คลายแล้วจะได้สายบิต 000100111 ขั้นตอนการคลายจะใช้อัลกอริทึมของ LZW แสดงใน [2]

2.4 ตัวดำเนินการทางพันธุกรรม

การสร้างประชากรรุ่นใหม่ทำได้โดยการใช้ตัวดำเนินการทางพันธุกรรม (genetic operator) [5] กับโครโมโซมของ LZWGA ที่ถูกเลือก ตัวดำเนินการทางพันธุกรรม ได้แก่

- การไขว้เปลี่ยน (crossover) เป็นการนำเอาโครโมโซมที่ถูกบีบอัดสองตัวมาผสมกันเพื่อให้ได้โครโมโซมตัวใหม่
- การกลายพันธุ์ (mutation) เป็นการสร้างโครโมโซมที่ถูกบีบอัดตัวใหม่โดยเลือกโครโมโซมที่ถูกบีบอัดรุ่นก่อนมาหนึ่งตัวและปรับเปลี่ยนส่วนใดส่วนหนึ่งของโครโมโซมอย่างสุ่ม

3. ปัญหา RoyalRoad และ RandomRoyalRoad

ปัญหา RoyalRoad เป็นปัญหาที่มีความสัมพันธ์ของบิตในโครโมโซม คำตอบที่ต้องการคือค่าของบิตทุกตัวในบล็อกมีค่าเท่ากับ 1 หากบิตใดบิตหนึ่งในบล็อกมีค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 แต่หากทุกบิตในบล็อกเป็น 1 หมด ค่าความเหมาะสมจะเท่ากับจำนวนบิตในบล็อกนั้น

ปัญหา RandomRoyalRoad ต่างจาก RoyalRoad ตรงที่คำตอบเกิดจากการสุ่ม ค่าของบิตในบล็อกที่เป็นคำตอบมีได้ทั้ง 0 และ 1 โครโมโซมที่ต้องการวัดค่าความเหมาะสมถ้ามีค่าของบิตในบล็อกทุกค่าเหมือนกับคำตอบ ค่าความเหมาะสมที่ได้จะเท่ากับจำนวนบิตในบล็อก แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

เหมือนกับคำตอบ บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันที เช่น

คำตอบ : 1101 1101 1101 1101

โครโมโซม : 1101 1110 1101 1101

ค่าความเหมาะสมที่ได้จะเท่ากับ 12 เนื่องจากในบล็อกที่ 2 บางบิตมีค่าไม่ตรงกับคำตอบ อัลกอริทึม RandomRoyalRoad มีการทำงาน 2 ส่วนคือ ส่วนของการสุ่มบล็อกและการวัดค่าความเหมาะสม [4]

4. เหตุผลที่เสนอตัวดำเนินการ Shift

จากการทดลองใช้ LZWGA แก้ปัญหา RoyalRoad โดยใช้โครโมโซม LZWGA ขนาด 47 ตำแหน่ง ซึ่งสามารถคลายได้ 1,128 บิตพอดี โครโมโซม LZWGA ที่เป็นคำตอบของปัญหาดังกล่าวเป็นตัวเลขที่เรียงกันตามลำดับ 1,2,3,... แต่คำตอบที่ LZWGA หาได้ยังเป็นตัวเลขที่เรียงกันไม่ถูกต้องคือมีบางตำแหน่งค่าซ้ำกัน ภาพที่ 2 แสดงคำตอบที่ LZWGA หาได้จากการทดลอง 3 ครั้ง ซึ่งไม่ใช่ตัวเลขเรียงกันในตำแหน่งที่ถูกต้อง ทำให้ไม่ใช่คำตอบที่ดีที่สุด

คำตอบ	1	2	3	4	5	6	7	8	9	10	11
ครั้งที่ 1	1	2	3	3	3	5	6	6	7	9	9
ครั้งที่ 2	0	1	2	3	4	5	6	7	8	9	10
ครั้งที่ 3	1	2	3	4	5	6	7	8	9	9	10

ภาพที่ 2 : คำตอบที่ LZWGA หาได้

คำตอบที่ LZWGA หาได้ เป็นคำตอบที่ใกล้จะเรียงกันได้อย่างถูกต้องแล้ว หากมีการเลื่อนตำแหน่งของโครโมโซมน่าจะทำให้โครโมโซมที่ได้มีค่าความเหมาะสมเพิ่มขึ้น ดังนั้นจึงเกิดแนวความคิดในการสร้างตัวดำเนินการ shift คือการเลื่อนตำแหน่งของโครโมโซมเพื่อปรับปรุงประสิทธิภาพการค้นหาของ LZWGA รายละเอียดของตัวดำเนินการ shift จะกล่าวถึงในส่วนถัดไป

5. ตัวดำเนินการ Shift สำหรับ LZWGA

5.1 Shift

การ shift คือการเลื่อนส่วนของโครโมโซมมาทางซ้ายหนึ่งตำแหน่ง ขั้นตอนเป็นดังนี้

(1) สุ่มเลือกตำแหน่งที่ i เพื่อ shift โดย $i < n$ และ n คือความยาวโครโมโซมของ LZWGA

(2) เริ่ม shift โดยเลื่อนโครโมโซมมาทางซ้าย 1 ตำแหน่ง ดังนี้

- แทนที่ตำแหน่งที่ i ด้วยตำแหน่งที่ i+1
- แทนที่ตำแหน่งที่ i+1 ด้วยตำแหน่งที่ i+2
- ...
- แทนที่ตำแหน่งที่ n–1 ด้วยตำแหน่งที่ n
- ตำแหน่งที่ n ให้คงค่าเดิมไว้

(3) เนื่องจากโครโมโซมของ LZWGA ไม่สามารถมีค่าเกินตำแหน่งของมันได้ เช่น ตำแหน่งที่ 2 ไม่สามารถมีค่าเกิน 2 ได้ (ค่าที่เป็นไปได้คือ {0, 1, 2}) การ shift อาจจะทำให้โครโมโซมอยู่ในตำแหน่งที่ไม่สามารถคลายได้ ดังนั้นโครโมโซมตำแหน่งที่เกิดการ shift จะต้องถูกนำมาผ่านเงื่อนไขดังสมการที่ 2

$$a_i = a_{i+1} \bmod (i+1) \qquad (2)$$

โดย a_i คือ ตำแหน่งที่ i ของโครโมโซมที่ถูกบีบอัดอยู่

ภาพที่ 3 แสดงตัวอย่างการ shift โครโมโซมที่ตำแหน่ง 1, 4 และ 7 ตามลำดับ โดยตำแหน่งของโครโมโซมที่เปลี่ยนไปส่งผลให้ค่าความเหมาะสมที่ได้เปลี่ยนแปลงไปด้วย

ต้นฉบับ	1	2	3	3	4	5	6	6	7	9	9
	ค่าความเหมาะสม : 44										
shift (1)	0	0	3	4	5	6	6	7	9	9	9
	ค่าความเหมาะสม : 0										
shift (4)	1	2	3	4	5	6	6	7	9	9	9
	ค่าความเหมาะสม : 56										
shift (7)	1	2	3	3	4	5	6	7	9	9	9
	ค่าความเหมาะสม : 44										

ภาพที่ 3 : การ shift โครโมโซม LZWGA ที่ตำแหน่งต่างๆ

5.2 Shift แบบมีเงื่อนไข (Shift and Check)

จากภาพที่ 3 จะเห็นว่าบางครั้งการ shift ไม่ทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้น เนื่องจากก่อน shift ตำแหน่งเดิมของโครโมโซมนั้นเหมาะสมอยู่แล้ว เช่น เมื่อ shift ตำแหน่งที่ 1 ค่าความเหมาะสมที่ได้จะเป็น 0 ดังนั้น จึงเกิดแนวความคิดที่จะปรับกระบวนการ shift โดยเพิ่มเงื่อนไขที่ต้องพิจารณาคือ ถ้า $a_{i+1} \bmod (i+1) = 0$ และ $a_{i+1} > 0$ แล้วเราจะเปลี่ยนให้ค่าของ a_i เป็นดังสมการที่ 3

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

$$a_i = a_{i+1} - 1 \quad (3)$$

สาเหตุที่นำค่าของ a_{i+1} ลบด้วย 1 ก็เพื่อหลีกเลี่ยงไม่ให้ค่าของโครโมโซมตำแหน่งที่ i มีค่าเป็น 0 ซึ่งจะทำให้ค่าความเหมาะสมของโครโมโซมที่ได้ลดลง สาเหตุที่ให้ลบด้วย 1 แทนที่จะลบด้วยเลขจำนวนเต็มอื่นก็เป็นเพราะจะทำให้ค่าความเหมาะสมจากการเลื่อนโครโมโซมมีโอกาสที่จะลดลงน้อยที่สุด

ภาพที่ 4 เปรียบเทียบการ shift และ shift and check ที่ตำแหน่งที่ 1 ของโครโมโซม พบว่าการ shift and check ทำให้ได้ค่าความเหมาะสมเพิ่มขึ้น

ภาพที่ 4 : การ shift และ shift and check โครโมโซมที่ตำแหน่งที่ 1

6. การทดลอง

การทดลองใช้ตัวดำเนินการ shift สำหรับ LZWGA ปัญหา RoyalRoad และ RandomRoyalRoad

6.1 ปัญหา RoyalRoad

ปัญหา RoyalRoad ขนาด 1,128 บิต บล็อก 4 บิต สามารถเข้ารหัสของคำตอบด้วย LZW ขนาด 47 ตำแหน่ง การทดลองแบ่งเป็น 2 กรณี คือ

(1) กำหนดโครโมโซม LZWGA ขนาด 47 ตำแหน่ง ซึ่งคลายได้เป็นโครโมโซม GA ขนาดเท่ากับคำตอบที่ต้องการพอดี และมีคำตอบในปริภูมิการค้นหาเพียงคำตอบเดียว (ซึ่ง LZWGA ธรรมดาหาไม่พบ)

(2) กำหนดโครโมโซม LZWGA ขนาด 50 ตำแหน่ง มีคำตอบในปริภูมิการค้นหาหลายคำตอบและ LZWGA มีโอกาสพบคำตอบมากขึ้น

6.2 ปัญหา RandomRoyalRoad

ปัญหา RandomRoyalRoad ขนาด 1,128 บิต บล็อก 4 บิต ใช้โครโมโซมของ LZWGA เท่ากันคือ 100 ตำแหน่ง แบ่งคำตอบที่ต้องการออกเป็น 2 กรณี คือ

(1) คำตอบคือ 1010 1010….1010 ซึ่งเป็นปัญหาที่มีแบบแผนสูง ค่อนข้างง่ายสำหรับ LZWGA

(2) คำตอบคือ 1101 1101….1101 ซึ่งเป็นปัญหาที่ยากขึ้น

6.3 พารามิเตอร์ที่ใช้ในการทดลอง

การทดลองแบ่งเป็น 3 วิธีคือ ไม่ใช้ตัวดำเนินการ shift, ใช้ตัวดำเนินการ shift, และใช้ตัวดำเนินการ shift and check กำหนดอัตราการ shift ตั้งแต่ 0.1 ถึง 1.0 เปรียบเทียบประสิทธิภาพของทั้ง 3 วิธี โดยวัดค่าความเหมาะสมของโครโมโซม ทดลองซ้ำปัญหาละ 20 ครั้ง

ตารางที่ 1 : พารามิเตอร์ที่ใช้ในการแก้ปัญหา RoyalRoad

พารามิเตอร์	กรณีที่ 1	กรณีที่ 2
จำนวนประชากร	1,000 ตัว	
ความยาวของโครโมโซม LZW	47	50
ค่าความเหมาะสมของคำตอบ	1,128	
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว	
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม	
อัตราการไขว้เปลี่ยน	80%	
อัตราการกลายพันธุ์	0%	
จำนวนรุ่นจำกัดไว้ที่	100 รุ่น	

ตารางที่ 2 : พารามิเตอร์ที่ใช้ในการแก้ปัญหา RandomRoyalRoad

พารามิเตอร์	RandomRoyalRoad
จำนวนประชากร	1,000 ตัว
ความยาวของโครโมโซม LZW	100
ค่าความเหมาะสมของคำตอบ	1,128
วิธีการคัดเลือก	แบบประลอง โครโมโซม 4 ตัว
วิธีการไขว้เปลี่ยน	การไขว้เปลี่ยนแบบยูนิฟอร์ม
อัตราการไขว้เปลี่ยน	80%
อัตราการกลายพันธุ์	0%
จำนวนรุ่นจำกัดไว้ที่	500 รุ่น

7. ผลการทดลอง

ปัญหาทั้งหมดที่นำมาทดลองมีดังนี้

(1) Royal Road ใช้โครโมโซม LZWGA 47 ตำแหน่ง

(2) Royal Road ใช้โครโมโซม LZWGA 50 ตำแหน่ง

(3) RandomRoyalRoad 1010..1010 ใช้โครโมโซม LZWGA 100 ตำแหน่ง

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

(4) RandomRoyalRoad 1101..1101 ใช้โครโมโซม LZWGA 100 ตำแหน่ง

จากการทดลองปรับอัตราการ shift ตั้งแต่ 0.1 ถึง 1.0 พบว่าอัตราการ shift ตั้งแต่ 0.5 ขึ้นไป การเปลี่ยนแปลงค่าความเหมาะสมมีไม่มากนัก ดังนั้นเราจึงพิจารณาการ shift ที่อัตรา 0.5 เพียงอย่างเดียว ภาพที่ 5 แสดงค่าความเหมาะสมของโครโมโซมจากการทดลองทั้ง 3 วิธี ซึ่ง LZWGA ที่ใช้ตัวดำเนินการ shift และ shift and check ได้โครโมโซมที่มีค่าความเหมาะสมมากกว่า LZWGA ธรรมดาเสมอ

ตารางที่ 3 : ค่าความเหมาะสมเฉลี่ยของการทดลองทั้ง 4 กรณี

	ไม่ shift	Shift 0.5	Shift and check 0.5
(1)	890.4	982.8 (ดีขึ้น 10.38 %)	1,119.8 (ดีขึ้น 23.34 %) พบคำตอบ 35 %
(2)	972.6	1,100.0 (ดีขึ้น 13.10 %) พบคำตอบ 35 %	1,128.0 (ดีขึ้น 14.13 %) พบคำตอบ 100 %
(3)	624.4	842.8 (ดีขึ้น 34.98 %)	966.0 (ดีขึ้น 40.53 %)
(4)	414.0	630.8 (ดีขึ้น 52.37 %)	749.2 (ดีขึ้น 53.14 %)

ค่าความเหมาะสมเฉลี่ยของการทดลองทั้ง 4 กรณี แสดงในตารางที่ 3 จะเห็นว่าตัวดำเนินการ shift และ shift and check ช่วยเพิ่มค่าความเหมาะสมของโครโมโซมได้เฉลี่ย 27.70% และ 32.78% ตามลำดับ

ภาพที่ 6 แสดงค่าความเหมาะสมของโครโมโซมในแต่ละช่วงเวลา จะเห็นว่าการ shift ทำให้ได้โครโมโซมที่มีค่าความเหมาะสมต่ำกว่า LZWGA ธรรมดาในช่วงแรกและได้ค่าความเหมาะสมสูงกว่า LZWGA ธรรมดาในช่วงหลัง ส่วนการ shift and check จะได้โครโมโซมที่มีค่าความเหมาะสมสูงกว่า LZWGA ธรรมดาเสมอ และสำหรับปัญหา RoyalRoad ที่ใช้ความยาวโครโมโซมของ LZWGA 50 ตำแหน่ง (ภาพ ข) ที่ LZWGA ธรรมดาหาคำตอบไม่พบ การใช้ตัวดำเนินการ shift and check จะทำให้หาคำตอบพบในเวลาเพียง 0.9 วินาที ดังนั้นตัวดำเนินการ shift และ shift and check จึงไม่ทำให้อัลกอริทึมเสียเวลาทำงานมากขึ้น แต่จะช่วยให้พบคำตอบได้เร็วขึ้นและได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้น

8. สรุป

บทความนี้นำเสนอตัวดำเนินการ shift 2 แบบ สำหรับ LZWGA คือ shift และ shift and check โดยทดลองกับปัญหา RoyalRoad และ RandomRoyalRoad ผลการทดลองพบว่าตัวดำเนินการ shift และ shift and check จะช่วยให้ได้โครโมโซม LZWGA ที่มีค่าความเหมาะสมสูงขึ้นเฉลี่ย 27.70% และ 32.78% ตามลำดับ อีกทั้งในกรณีที่จำกัดความยาวโครโมโซม ซึ่ง LZWGA ธรรมดาหาคำตอบไม่พบ ตัวดำเนินการ shift จะช่วยให้สามารถพบคำตอบได้

สำหรับการหาคำตอบที่ไม่มีแบบแผน (เช่น ปัญหา RandomRoyalRoad ที่มีคำตอบเป็นบล็อก 1101) ถึงแม้ตัวดำเนินการ shift จะช่วยให้ได้โครโมโซมที่มีค่าความเหมาะสมมากขึ้นแต่ก็ยังหาคำตอบไม่พบ ซึ่งส่วนนี้ผู้วิจัยจะศึกษาหาสาเหตุและวิธีการแก้ไขต่อไป

9. เอกสารอ้างอิง

[1] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Profcssional, USA, 1989.

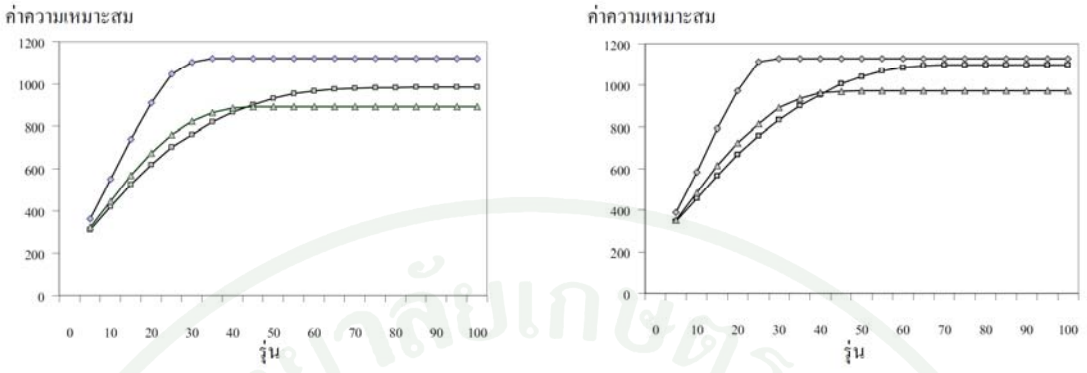
[2] นริศ กุณาศล, วรเศรษฐ สุวรรณิก, และ ประภาส จงสถิตย์วัฒนา, "การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม," *Proceedings of Electrical Engineering Conference (EECON-28)*, หน้า 861-864, 20-21 ตุลาคม, 2548.

[3] Naris Kunasol, Worasait Suwannik, and Prabhas Chongstitvatana, "Solving One-Million-Bit Problems Using LZWGA," *Proceedings of International Symposium on Communications and Information Technologies (ISCIT)*, pp. 32-36, October 18-20, 2006

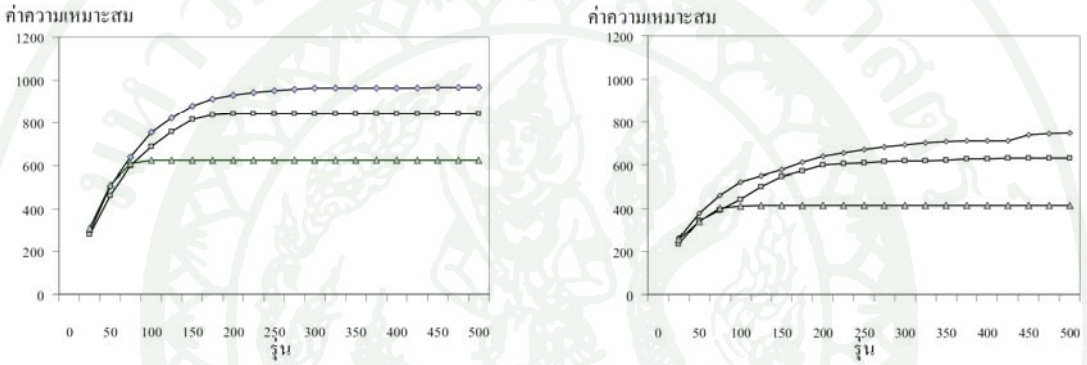
[4] สาวิตรี ฉ่ำลมัย และ วรเศรษฐ สุวรรณิก, "ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด," *Proceedings of Electrical Engineering Conference (EECON-30)*, หน้า 653-656, 25-26 ตุลาคม, 2550.

[5] วรเศรษฐ สุวรรณิก, นริศ กุณาศล, และ ประภาส จงสถิตย์วัฒนา, "ขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด," *Proceedings of Northeastern Computer Science and Engineering Conference (NECSEC)*, หน้า 203-211, 31 มีนาคม - 1 เมษายน, 2548.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



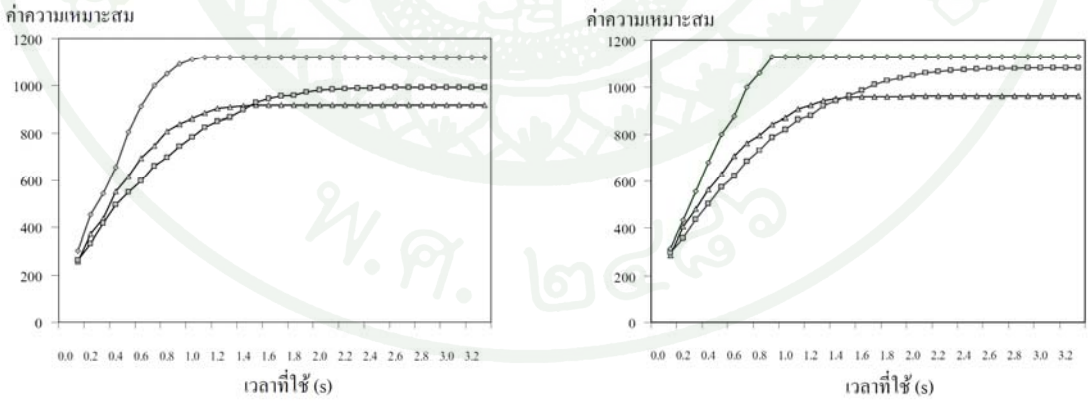
(ก) RoyalRoad ใช้โครโมโซม LZWGA 47 ตำแหน่ง (ข) Royal Road ใช้โครโมโซม LZWGA 50 ตำแหน่ง



(ค) RandomRoyalRoad คำตอบที่ต้องการคือ 1010 1010....1010 ใช้โครโมโซม LZWGA 100 ตำแหน่ง

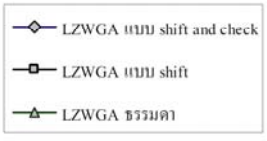
(ง) RandomRoyalRoad คำตอบที่ต้องการคือ 1101 1101....1101 ใช้โครโมโซม LZWGA 100 ตำแหน่ง

ภาพที่ 5 : ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift and check ในแต่ละรุ่น



(ก) RoyalRoad ใช้โครโมโซม LZWGA 47 ตำแหน่ง (ข) Royal Road ใช้โครโมโซม LZWGA 50 ตำแหน่ง

ภาพที่ 6 : ค่าความเหมาะสมของคำตอบที่ได้จาก LZWGA ธรรมดา, LZWGA แบบ shift และ LZWGA แบบ shift and check ในแต่ละช่วงเวลา



ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

การวิเคราะห์ประสิทธิภาพของตัวดำเนินการ Shift-and-Check ใน LZWGA
A Performance Analysis of a Shift-and-Check Operator in LZWGA

อณิมา รอดเสียงล้ำ และ วรเศรษฐ สุวรรณิก
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
50 ถ.พหลโยธิน แขวงลาดยาว เขตจตุจักร กรุงเทพฯ 10900 โทรศัพท์ : 0-2562-5555 E-mail: g5164149@ku.ac.th, worasait.suwannik@gmail.com

บทคัดย่อ

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) LZWGA เข้ารหัสโครโมโซมในรูปแบบของการบีบอัด อัลกอริทึมนี้สามารถใช้ตัวดำเนินการทางพันธุกรรมแบบเดียวกับ GA คือ การไขว้เปลี่ยนและการกลายพันธุ์ ในการวิจัยก่อนหน้านี้เราได้เสนอตัวดำเนินการ shift-and-check ที่ช่วยเพิ่มประสิทธิภาพให้กับ LZWGA สำหรับในบทความนี้ เราเสนอการวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการดังกล่าว ผลการวิเคราะห์พบว่าตัวดำเนินการ shift-and-check สามารถสร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้นหรือเท่าเดิมด้วยความน่าจะเป็นมากกว่า 0.5

คำสำคัญ: การเข้ารหัสแบบ LZW, ขั้นตอนวิธีเชิงพันธุกรรม, ตัวดำเนินการ shift-and-check

Abstract

LZW encoding in Genetic Algorithm (LZWGA) is an algorithm developed from Genetic Algorithm (GA). LZWGA encodes a chromosome in a compressed format. This algorithm can use the same genetic operators as GA, which are crossover and mutation. In our previous work, a shift-and-check operator was proposed to improve the performance of LZWGA. In this paper, we present an analysis of the operator. The result shows that the shift-and-check operator can produce a chromosome whose fitness is higher than or equal to the original one with probability greater than 0.5.

Keywords: LZWGA, GA, Shift-and-check operator

1. บทนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm - GA) เป็นวิธีการค้นหาคำตอบที่เลียนแบบวิวัฒนาการทางธรรมชาติ [1] คำตอบแต่ละตัวจะถูกเข้ารหัสอยู่ในรูปของโครโมโซม (chromosome) ที่เป็นเลขฐานสอง เมื่อโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหาก็จะมีขนาดใหญ่ขึ้นตามไปด้วย เช่นปัญหาขนาด 128 บิต มีคำตอบที่เป็นไปได้ 2^{128} แบบ ในขณะที่ปัญหาขนาดหนึ่งล้านบิต จะมีคำตอบที่เป็นไปได้ทั้งหมด $2^{1000000}$ แบบ

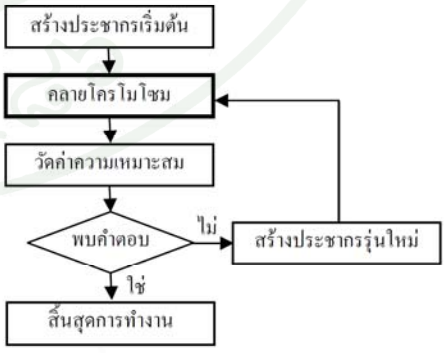
การเข้ารหัสแบบ LZWในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) ใช้การเข้ารหัสโครโมโซมแบบบีบอัดโดยใช้อัลกอริทึมการบีบอัดของ Lempel-Ziv-Welch ทำให้โครโมโซมมีขนาดเล็กลง [2] การวิวัฒนาการทำกับข้อมูลที่ถูกบีบอัดซึ่งเป็นอาร์เรย์ของเลขจำนวนเต็ม ก่อนที่จะวัดค่าความเหมาะสม เราต้องนำโครโมโซมมาผ่านกระบวนการคลายโดยใช้อัลกอริทึมการคลายของ LZW เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำมาคลายจะได้โครโมโซมของ GA คือ 1010101 แล้วจึงนำไปวัดค่าความเหมาะสม (fitness) ต่อไป

การทำงานของ LZWGA คล้ายกับ GA คือประกอบด้วยตัวดำเนินการทางพันธุกรรมที่สำคัญได้แก่ การไขว้เปลี่ยน, และการกลายพันธุ์ นอกจากนี้ LZWGA ยังมีตัวดำเนินการ shift-and-check (SC) ที่ช่วยทำให้ LZWGA พบคำตอบได้เร็วขึ้นและได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้น [3] บทความนี้เสนอการวิเคราะห์ประสิทธิภาพการทำงานของตัวดำเนินการ SC ว่าหลังจากการ SC แล้ว มีความน่าจะเป็นที่จะได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นเพียงใด โดยการวิเคราะห์แสดงในส่วนที่ 5

2. LZWGA

2.1 ขั้นตอนการทำงาน

ขั้นตอนการทำงานของ LZWGA คล้ายกับ GA แต่จะ เพิ่มขั้นตอนการคลายโครโมโซมก่อนที่จะนำมาวัดค่าความเหมาะสม ดังแสดงในรูปที่ 1



รูปที่ 1 ขั้นตอนการทำงานของ LZWGA

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

2.2 โครโมโซม

โครโมโซมของ GA เป็นเลขฐานสอง ส่วนโครโมโซมที่ถูกบีบอัดของ LZWGA เป็นอาร์เรย์ของเลขจำนวนเต็มตั้งแต่ 0-n โดย n คือ ขนาดของโครโมโซมที่ถูกบีบอัด การสร้างโครโมโซมในประชากรรุ่นแรกของ LZWGA ต้องสุ่มให้อยู่ในเงื่อนไขที่สามารถนำไปคลายได้ ซึ่งก็คือโครโมโซมตำแหน่งต่างๆ จะมีค่าได้ไม่เกินตำแหน่งนั้น ตัวอย่างเช่น โครโมโซมตำแหน่งที่ 1 มีค่าที่เป็นไปได้คือ {0, 1} โครโมโซมตำแหน่งที่ 2 มีค่าที่เป็นไปได้คือ {0, 1, 2} ซึ่งสามารถสรุปได้ดังสมการที่ 1

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

เมื่อ a_i คือ ตำแหน่งที่ i ของโครโมโซมที่ถูกบีบอัดอยู่
r คือ เลขที่สุ่มขึ้นมา
n คือ ขนาดของโครโมโซม

2.3 การคลายและการวัดค่าความเหมาะสม

โครโมโซมของ LZWGA อยู่ในรูปของการบีบอัด ดังนั้นก่อนวัดค่าความเหมาะสม จึงต้องมีการคลายโครโมโซมก่อน ตัวอย่างเช่น โครโมโซมของ LZWGA คือ {0, 2, 1, 3, 1, 1} คลายแล้วจะได้สายบิต 000100111 ขั้นตอนการคลายใช้อัลกอริทึมของ LZW แสดงใน [2] สำหรับการวัดค่าความเหมาะสมจะทำแบบเดียวกับ GA

2.4 ตัวดำเนินการทางพันธุกรรมของ LZWGA

การสร้างประชากรรุ่นใหม่ทำได้โดยการใช้ตัวดำเนินการทางพันธุกรรม (genetic operator) กับโครโมโซมของ LZWGA ที่ถูกเลือกขึ้นมา ตัวดำเนินการทางพันธุกรรม ได้แก่ การไขว้เปลี่ยน, การกลายพันธุ์, และการ shift

(1) การไขว้เปลี่ยน (Crossover)

การไขว้เปลี่ยนเป็นการนำเอาโครโมโซมที่ถูกบีบอัดสองตัวมาผสมกันเพื่อให้ได้โครโมโซมตัวใหม่ การไขว้เปลี่ยนโดยทั่วไปได้แก่

- การไขว้เปลี่ยนแบบจุดเดียว (One-Point Crossover) เป็นการสุ่มจุดตัดเพียงจุดเดียวและไขว้เปลี่ยนโครโมโซมที่จุดตัดนั้น
- การไขว้เปลี่ยนแบบสองจุด (Two-Point Crossover) เป็นการสุ่มจุดตัดสองจุดแล้วไขว้เปลี่ยนโครโมโซมระหว่างจุดตัดทั้งสอง
- การไขว้เปลี่ยนแบบยูนิฟอร์ม (Uniform Crossover) เป็นการแลกเปลี่ยนโครโมโซมเฉพาะจุด

(2) การกลายพันธุ์ (Mutation)

การกลายพันธุ์เป็นการสร้างโครโมโซมที่ถูกบีบอัดตัวใหม่โดยเลือกโครโมโซมที่ถูกบีบอัดรุ่นก่อนมาหนึ่งตัวและปรับเปลี่ยนส่วนใดส่วนหนึ่งของโครโมโซมอย่างสุ่ม

(3) การ Shift

การ shift เป็นการสร้างโครโมโซมรุ่นใหม่โดยเลื่อนตำแหน่งของโครโมโซมหลังตำแหน่งที่สุ่มได้ไปทางซ้ายหนึ่งตำแหน่ง รายละเอียดของตัวดำเนินการ shift แสดงในส่วนที่ 4

3. ปัญหาสำหรับทดสอบ

ปัญหาที่ใช้ในการทดสอบครั้งนี้มี 4 ปัญหา คือ OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad

3.1 OneMax (OM)

ปัญหา OneMax คือ ปัญหาการนับจำนวนเลขหนึ่งในโครโมโซม ค่าความเหมาะสมที่ได้ก็คือ จำนวนเลขหนึ่งทั้งหมดที่พบในโครโมโซม ดังนั้นโครโมโซมที่ดีที่สุดคือโครโมโซมที่ทุกบิตมีค่าเป็นหนึ่ง ปัญหา OneMax จำลองปัญหาการหาค่าที่เหมาะสมที่สุด (optimization problem) ที่ตัวแปรทุกตัวเป็นอิสระต่อกัน

3.2 RandomMax (RM)

ปัญหา RandomMax [4] คล้ายกับ OneMax เพราะเป็นปัญหาการนับจำนวนบิตที่ตรงกับคำตอบ แต่ต่างกันตรงที่คำตอบเกิดจากการสุ่ม โดยค่าความเหมาะสมที่ได้คือ จำนวนบิต ณ ตำแหน่งเดียวกับคำตอบที่มีค่าเท่ากัน เช่น

คำตอบที่สุ่มมา : 10100101
โครโมโซม : 10000100

ค่าความเหมาะสมที่ได้จะเป็นจำนวนบิตที่มีค่าตรงกับคำตอบ ซึ่งมีค่าเท่ากับ 6 เนื่องจากในบิตที่ 3 และ 8 (จากซ้าย) มีค่าไม่ตรงกับคำตอบที่สุ่มมา

3.3 RoyalRoad (RR)

ปัญหา RoyalRoad เป็นปัญหาที่จำลองความสัมพันธ์ของบิตในโครโมโซม โครโมโซมถูกแบ่งเป็นบล็อก คำตอบที่ต้องการคือทุกบิตในบล็อกมีค่าเท่ากับ 1 หากบิตใดบิตหนึ่งในบล็อกมีค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 แต่หากทุกบิตในบล็อกเป็น 1 หมด ค่าความเหมาะสมจะเท่ากับจำนวนบิตในบล็อกนั้น

3.4 RandomRoyalRoad (RRR)

ปัญหา RandomRoyalRoad [4] คล้ายกับ RoyalRoad แต่ต่างกันตรงที่คำตอบของแต่ละบล็อกเกิดจากการสุ่ม ค่าของบิตในบล็อกที่เป็นคำตอบจึงมีได้ทั้ง 0 และ 1 ถ้าบิตในบล็อกของโครโมโซมที่ต้องการวัดค่าความเหมาะสมมีค่าเหมือนกับคำตอบ ค่าความเหมาะสมที่ได้จะเท่ากับจำนวนบิตในบล็อกนั้น แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่เหมือนกับคำตอบ บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันที เช่น

คำตอบ : 1101 1101 1101 1101
โครโมโซม : 1101 1110 1101 1101

เนื่องจากในบล็อกที่ 1, 3, และ 4 มีค่าตรงกับคำตอบ แต่ในบล็อกที่ 2 มีบางบิตมีค่าไม่ตรงกับคำตอบ ดังนั้นค่าความเหมาะสมที่ได้จึงมีค่าเท่ากับ 12

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

4. ตัวดำเนินการ Shift และ Shift-and-Check

4.1 การทำงานของตัวดำเนินการ Shift และ Shift-and-Check

(1) Shift

การ shift เป็นการเลื่อนโครโมโซมหลังตำแหน่งที่สุ่มได้ไปทางซ้าย 1 ตำแหน่ง ขั้นตอนการทำงานของตัวดำเนินการ shift แสดงใน [3]

(2) Shift-and-Check (SC)

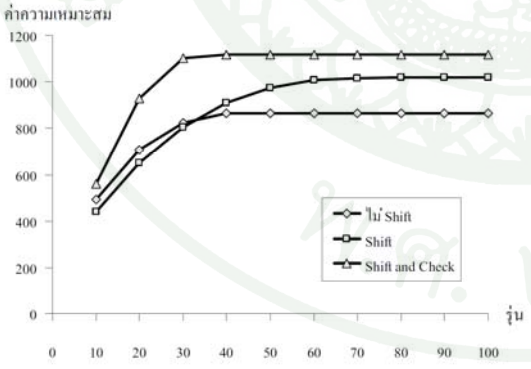
เนื่องจากบางครั้งการ shift ไม่ช่วยให้โครโมโซมที่ได้มีค่าความเหมาะสมสูงขึ้น เพราะก่อน shift ตำแหน่งเดิมของโครโมโซมนั้นเหมาะสมอยู่แล้ว ดังนั้นกระบวนการ shift จึงถูกปรับปรุงเป็น shift-and-check โดยเพิ่มเงื่อนไขคือ ถ้า $a_{i+1} \bmod (i+1) = 0$ โดยที่ $a_{i+1} \neq 0$ แล้ว ค่าของ a_i จะเป็นดังสมการที่ 2

$$a_i = a_{i+1} - 1 \qquad (2)$$

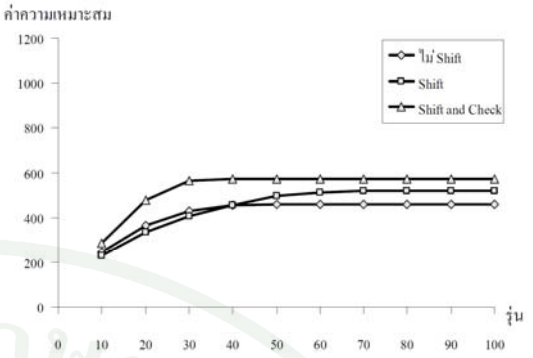
4.2 การ Shift กับปัญหา OneMax และ RandomMax

การใช้ shift และ SC แก้ปัญหา RoyalRoad และ RandomRoyalRoad ทำให้โครโมโซมที่ได้มีค่าความเหมาะสมสูงขึ้น [3] บทความนี้เพิ่มการทดลองกับปัญหา OneMax และ RandomMax โดยใช้ความยาวโครโมโซม LZWGA 47 ตำแหน่ง จำนวนรุ่นสูงสุด 100 รุ่นและใช้พารามิเตอร์อื่นๆ เหมือนใน [3] ผลการทดลองพบว่า shift และ SC ช่วยทำให้ได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นเช่นกัน ดังแสดงในรูปที่ 2

บทความนี้เราจะวิเคราะห์ประสิทธิภาพของตัวดำเนินการ SC เพียงอย่างเดียว โดยการทดลองเพื่อวิเคราะห์ประสิทธิภาพแสดงในส่วนถัดไป



(ก) ปัญหา OneMax



(ข) ปัญหา RandomMax

รูปที่ 2 LZWGA ที่ไม่ใช้ shift, ใช้ shift ,และ ใช้ SC

5. การวิเคราะห์

การวิเคราะห์ประสิทธิภาพของตัวดำเนินการ SC โดยกำหนดความยาวโครโมโซมของ LZWGA ตั้งแต่ 4 ถึง 8 ตำแหน่ง มีขั้นตอนดังนี้

(1) สร้างโครโมโซม LZWGA ทุกรูปแบบที่เป็นไปได้

(2) วัดค่าความเหมาะสมของโครโมโซมทุกตัว

(3) นำโครโมโซมแต่ละตัวมาผ่านการ SC ทีละตำแหน่ง เริ่มตั้งแต่ตำแหน่งแรกจนครบทุกตำแหน่งและวัดค่าความเหมาะสมที่ได้จากการ SC แต่ละครั้ง

(4) เปรียบเทียบค่าความเหมาะสมของโครโมโซมที่ได้หลังการ SC เทียบกับค่าความเหมาะสมของโครโมโซมต้นฉบับ

(5) คำนวณความน่าจะเป็นที่ SC ทำให้โครโมโซมของ LZWGA มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมโดยใช้สมการที่ 3

$$P = \frac{NC}{TC} \qquad (3)$$

โดยที่

P คือ ความน่าจะเป็นที่ค่าความเหมาะสมของโครโมโซมเพิ่มขึ้นหรือเท่าเดิม

NC คือ จำนวนโครโมโซมที่เมื่อใช้ SC แล้วค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม

TC คือ จำนวนโครโมโซมทุกรูปแบบที่เป็นไปได้และนำมาผ่านการ SC ทุกตำแหน่ง ค่าของ TC คำนวณโดยใช้สมการที่ 4

$$TC = l \times (l+1)! \qquad (4)$$

เมื่อ l คือ ความยาวโครโมโซมของ LZWGA

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

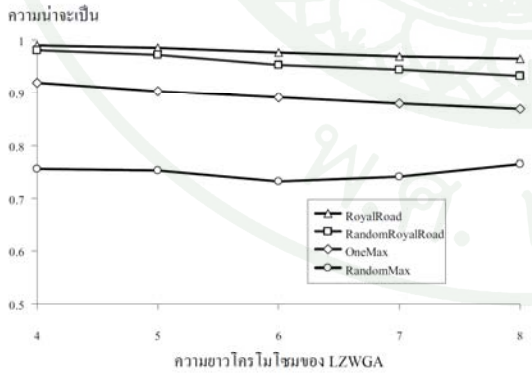
6. ผลการวิเคราะห์

ผลการวิเคราะห์ประสิทธิภาพของตัวดำเนินการ SC กับปัญหาต่างๆ พบว่าตัวดำเนินการ SC สร้างโครโมโซมที่ส่วนใหญ่มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม ดังแสดงในตารางที่ 1 เมื่อกำหนดโครโมโซม LZWGA ขนาด 4 ตำแหน่ง พบว่าถ้านำมา SC ทุกตำแหน่งจะทำให้เกิดโครโมโซมที่เป็นไปได้ทั้งหมด 480 แบบ โดยจะมีโครโมโซมที่มีค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมถึง 441 แบบในปัญหา OneMax, 363 แบบในปัญหา RandomMax, 475 แบบในปัญหา RoyalRoad และ 471 แบบในปัญหา RandomRoyalRoad

ตารางที่ 1 จำนวนโครโมโซมที่ SC แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิมในปัญหาทดสอบ (l คือความยาวโครโมโซมของ LZWGA)

l	จำนวนโครโมโซมทั้งหมด	จำนวนโครโมโซมที่ SC แล้วได้ค่าความเหมาะสมเพิ่มขึ้นหรือเท่าเดิม			
		OM	RM	RR	RRR
4	480	441	363	475	471
5	3,600	3,252	2,715	3,548	3,497
6	30,240	26,928	22,174	29,540	28,825
7	282,240	248,157	209,327	273,577	266,326
8	2,903,040	2,523,477	2,220,282	2,802,036	2,704,113

เมื่อนำข้อมูลจากตารางที่ 1 มาคำนวณค่าความน่าจะเป็นแล้ววาดกราฟจะได้ดังรูปที่ 3 ซึ่งแสดงค่าความน่าจะเป็นที่ตัวดำเนินการ SC สร้างโครโมโซมที่มีค่าความเหมาะสมสูงขึ้นหรือเท่าเดิมในปัญหาต่างๆ ที่ความยาวโครโมโซมตั้งแต่ 4 ถึง 8 ตำแหน่ง เมื่อโครโมโซม LZWGA มีความยาวมากขึ้น ความน่าจะเป็นที่เมื่อ SC แล้วได้ค่าความเหมาะสมเพิ่มขึ้นจะค่อยๆ ลดลง แต่ในปัญหา RandomMax ซึ่งคำตอบเกิดจากการสุ่ม ความยาวโครโมโซมไม่มีผลต่อความน่าจะเป็นที่การ SC แล้วจะได้คำตอบที่ดีขึ้นหรือเท่าเดิม



รูปที่ 3 ความน่าจะเป็นที่การ SC จะทำให้โครโมโซมมีค่าความเหมาะสมสูงขึ้นหรือเท่าเดิม

7. สรุปผล

ในการทดสอบกับปัญหา OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad ตัวดำเนินการ SC สามารถสร้างโครโมโซมที่ดีขึ้นหรือเท่ากับโครโมโซมเดิมด้วยความน่าจะเป็นเฉลี่ยเท่ากับ 0.89, 0.75, 0.97, และ 0.95 ตามลำดับ เนื่องจากความน่าจะเป็นดังกล่าวมีมากกว่า 0.5 จึงส่งผลให้ LZWGA พบคำตอบเร็วขึ้นหรือไม่ก็ได้โครโมโซมที่มีค่าความเหมาะสมสูงขึ้นเมื่อใช้ตัวดำเนินการ SC

เนื่องจากในการบีบอัดแบบ LZW ตำแหน่งของอินพุตมีผลต่อเอาต์พุตของการคลาย ดังนั้นสิ่งที่ต้องศึกษาต่อไปอาจจะเป็นการวิเคราะห์ว่า SC ช่วยในการจัดเรียงคำตอบให้ตรงกับตำแหน่งได้ถูกต้องมากหรือน้อยเพียงใด ซึ่งผลการวิเคราะห์ดังกล่าวอาจจะนำไปสู่การพัฒนาตัวดำเนินการตัวใหม่ที่เพิ่มประสิทธิภาพให้กับ LZWGA มากยิ่งขึ้นไปอีก

เอกสารอ้างอิง

[1] D.E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning," Addison-Wesley Professional, Canada, 1989.

[2] นริศ กุณาศล, วรเศรษฐ สุวรรณิก และ ประภาส จงสถิตย์วัฒนา, "การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม," Proceedings of Electrical Engineering Conference (EECON-28), หน้า 861-864, 2548.

[3] อณิมา รอดเสียงล้ำ และ วรเศรษฐ สุวรรณิก, "ตัวดำเนินการ Shift สำหรับ LZWGA," The 5th National Conference on Computing and Information Technology (NCCIT), หน้า 320-325, 2552.

[4] สาวิตรี ฉ่ำลมัย และ วรเศรษฐ สุวรรณิก, "ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด," Proceedings of Electrical Engineering Conference (EECON-30), หน้า 653-656, 2550.

นางสาวอณิมา รอดเสียงล้ำ
การศึกษา : วิทยาศาสตรบัณฑิต
มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา
ปัจจุบัน : ศึกษาระดับปริญญาโท
สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
งานวิจัยที่สนใจ : ขั้นตอนวิธีเชิงพันธุกรรม

ดร.วรเศรษฐ สุวรรณิก
การศึกษา : วิศวกรรมศาสตรดุษฎีบัณฑิต
จุฬาลงกรณ์มหาวิทยาลัย
ปัจจุบัน : อาจารย์ประจำ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
งานวิจัยที่สนใจ : ขั้นตอนวิธีเชิงพันธุกรรม

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

การเปรียบเทียบประสิทธิภาพการทำงานของ LZWGA บนวิธีการคัดเลือกที่แตกต่าง
Comparing the Performance of LZWGA with Various Selection Methods

อณิมา รอดเสียงล้ำ และ วรเศรษฐ สุวรรณิก
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
50 ถ.พหลโยธิน แขวงลาดยาว เขตจตุจักร กรุงเทพฯ 10900 โทรศัพท์ : 0-2562-5555 E-mail: g5164149@ku.ac.th, worasait.suwannik@gmail.com

บทคัดย่อ

การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) โดยใช้การเข้ารหัสโครโมโซมในรูปของการบีบอัด การทำงานของ LZWGA ประกอบด้วยกระบวนการทางพันธุกรรมเช่นเดียวกับ GA คือ การคัดเลือก, การไขว้เปลี่ยน, และการกลายพันธุ์ บทความนี้เสนอการเปรียบเทียบประสิทธิภาพการทำงานของ LZWGA เมื่อใช้วิธีการคัดเลือกที่แตกต่างกัน ผลการทดลองพบว่าวิธีการคัดเลือกที่ทำให้ LZWGA ได้โครโมโซมที่มีค่าความเหมาะสมมากที่สุดคือการคัดเลือกแบบประลอง

คำสำคัญ: การเข้ารหัสแบบ LZW, ขั้นตอนวิธีเชิงพันธุกรรม, การคัดเลือก

Abstract

LZW encoding in Genetic Algorithm (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. Similar to GA, LZWGA consists of selection, crossover, and mutation. This paper compares the performance of LZWGA with various selection and crossover methods. The result shows that LZWGA is best performed using tournament selection.

Keywords: LZWGA, GA, Selection

1. บทนำ

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm - GA) เป็นวิธีการค้นหาคำตอบที่เลียนแบบวิวัฒนาการทางธรรมชาติ [1] คำตอบแต่ละตัวจะถูกเข้ารหัสอยู่ในรูปของโครโมโซม (chromosome) ที่เป็นเลขฐานสอง เมื่อโครโมโซมมีขนาดใหญ่ขึ้น ปริภูมิการค้นหาก็จะมีขนาดใหญ่ขึ้นตามไปด้วย เช่นปัญหาขนาด 128 บิต มีคำตอบที่เป็นไปได้ 2^{128} แบบ ในขณะที่ปัญหาขนาดหนึ่งล้านบิต จะมีคำตอบที่เป็นไปได้ทั้งหมด $2^{1000000}$ แบบ

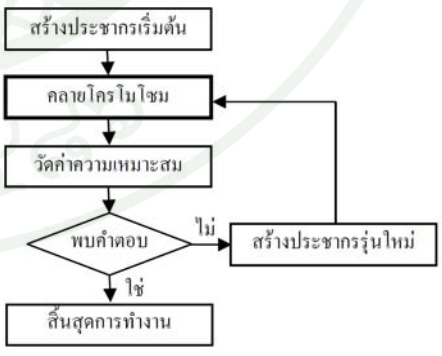
การเข้ารหัสแบบ LZWในขั้นตอนวิธีเชิงพันธุกรรม (LZWGA) ใช้การเข้ารหัสโครโมโซมแบบบีบอัดโดยใช้อัลกอริทึมการบีบอัดของ Lempel-Ziv-Welch ทำให้โครโมโซมมีขนาดเล็กลง [3] ช่วยลดขนาดของปริภูมิการค้นหา การวิวัฒนาการทำกับข้อมูลที่ถูกบีบอัดซึ่งเป็นอาร์เรย์ของเลขจำนวนเต็ม ก่อนที่จะวัดค่าความเหมาะสม เราต้องนำโครโมโซมมาผ่านกระบวนการคลายโดยใช้อัลกอริทึมการคลายของ LZW เช่น ถ้าโครโมโซมของ LZWGA คือ {1, 0, 2, 4} เมื่อนำมาคลายจะได้โครโมโซมของ GA คือ 1010101 แล้วจึงนำไปวัดค่าความเหมาะสม (fitness) การทำงานของ LZWGA คล้ายกับ GA คือประกอบด้วยการดำเนินการทางพันธุกรรมที่สำคัญได้แก่ การคัดเลือก, การไขว้เปลี่ยน, และการกลายพันธุ์

การคัดเลือกเป็นกระบวนการสำคัญสำหรับการสร้างประชากรรุ่นใหม่ใน GA วิธีการคัดเลือกมีผลต่อประสิทธิภาพการทำงานของ GA โดยการคัดเลือกแบบประลองทำให้ GA ทำงานได้อย่างมีประสิทธิภาพมากที่สุด [4] อย่างไรก็ดี ไม่เคยมีการศึกษามาก่อนว่าวิธีการคัดเลือกจะมีผลต่อประสิทธิภาพการทำงานของ LZWGA มากหรือน้อยเพียงใด บทความนี้เสนอการเปรียบเทียบประสิทธิภาพการทำงานของ LZWGA เมื่อใช้วิธีการคัดเลือกที่ต่างกัน 3 วิธีคือ การคัดเลือกแบบประลอง, การคัดเลือกแบบวงล้อรูเล็ตต์, และการคัดเลือกแบบจัดลำดับ

2. LZWGA

2.1 ขั้นตอนการทำงาน

การทำงานของ LZWGA คล้ายกับการทำงานของ GA แต่จะเพิ่มขั้นตอนการคลายโครโมโซมก่อนที่จะนำมาวัดค่าความเหมาะสม ดังแสดงในรูปที่ 1



รูปที่ 1 ขั้นตอนการทำงานของ LZWGA

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

2.2 โครโมโซม

โครโมโซมของ GA เป็นเลขฐานสอง ส่วนโครโมโซมที่ถูกบีบอัดของ LZWGA เป็นอาร์เรย์ของเลขจำนวนเต็มตั้งแต่ 0-n โดย n คือขนาดของโครโมโซมที่ถูกบีบอัด การสร้างโครโมโซมในประชากรรุ่นแรกของ LZWGA ต้องสุ่มให้อยู่ในเงื่อนไขที่สามารถนำไปคลายได้ ซึ่งก็คือโครโมโซมตำแหน่งต่างๆ จะมีค่าได้ไม่เกินตำแหน่งนั้น ตัวอย่างเช่น โครโมโซมตำแหน่งที่ 1 มีค่าที่เป็นไปได้คือ {0, 1} โครโมโซมตำแหน่งที่ 2 มีค่าที่เป็นไปได้คือ {0, 1, 2} ซึ่งสามารถสรุปได้ดังสมการที่ 1

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

เมื่อ a_i คือ ตำแหน่งที่ i ของโครโมโซมที่ถูกบีบอัดอยู่
r คือ เลขที่สุ่มขึ้นมา
n คือ ขนาดของโครโมโซม

2.3 การคลายและการวัดค่าความเหมาะสม

โครโมโซมของ LZWGA อยู่ในรูปของการบีบอัด ดังนั้นก่อนวัดค่าความเหมาะสม จึงต้องมีการคลายโครโมโซมก่อน ตัวอย่างเช่น โครโมโซมของ LZWGA คือ {0, 2, 1, 3, 1, 1} คลายแล้วจะได้สายบิต 000100111 ขั้นตอนการคลายใช้อัลกอริทึมของ LZW แสดงใน [3]

2.4 ตัวดำเนินการทางพันธุกรรมของ LZWGA

การสร้างประชากรรุ่นใหม่ทำได้โดยการใช้ตัวดำเนินการทางพันธุกรรม (genetic operator) กับโครโมโซมของ LZWGA ที่ถูกเลือกขึ้นมา ตัวดำเนินการทางพันธุกรรม ได้แก่ การไขว้เปลี่ยน, การกลายพันธุ์, และการ Shift

(1) การไขว้เปลี่ยน (Crossover)

การไขว้เปลี่ยนเป็นการนำเอาโครโมโซมที่ถูกบีบอัดสองตัวมาผสมกันเพื่อให้ได้โครโมโซมตัวใหม่ การไขว้เปลี่ยนที่ใช้กันทั่วไปได้แก่

- การไขว้เปลี่ยนแบบจุดเดียว (One-Point Crossover) เป็นการสุ่มจุดตัดเพียงจุดเดียวและไขว้เปลี่ยนโครโมโซมที่จุดตัดนั้น
- การไขว้เปลี่ยนแบบสองจุด (Two-Point Crossover) เป็นการสุ่มจุดตัดสองจุดแล้วไขว้เปลี่ยนโครโมโซมระหว่างจุดตัดทั้งสอง
- การไขว้เปลี่ยนแบบยูนิฟอร์ม (Uniform Crossover) เป็นการแลกเปลี่ยนโครโมโซมเฉพาะจุด

(2) การกลายพันธุ์ (Mutation)

การกลายพันธุ์เป็นการสร้างโครโมโซมที่ถูกบีบอัดตัวใหม่โดยเลือกโครโมโซมที่ถูกบีบอัดรุ่นก่อนมาหนึ่งตัวและปรับเปลี่ยนส่วนใดส่วนหนึ่งของโครโมโซมอย่างสุ่ม

(3) การ Shift

การ shift มี 2 แบบคือ shift และ shift and check โดยเป็นการเลื่อนโครโมโซมหลังตำแหน่งที่สุ่มได้ไปทางซ้าย 1 ตำแหน่ง รายละเอียดการทำงานของตัวดำเนินการ shift และ shift and check แสดงใน [5]

3. วิธีการคัดเลือก

เมื่อต้องการสร้างประชากรรุ่นใหม่จะต้องเลือกโครโมโซมพ่อแม่ วิธีการคัดเลือกที่ถูกนำมาเปรียบเทียบในบทความนี้มี 3 วิธี คือ การคัดเลือกแบบประลอง, แบบวงล้อรูเล็ตต์, และแบบจัดลำดับ

3.1 การคัดเลือกแบบประลอง (Tournament Selection)

การคัดเลือกแบบประลองจะสุ่มเลือกโครโมโซมเป็นจำนวนเท่ากับขนาดของการประลอง (tournament size) แล้วนำมาเปรียบเทียบค่าความเหมาะสม โครโมโซมที่มีค่าความเหมาะสมสูงที่สุดจะเป็นตัวที่ถูกเลือกเพื่อนำไปใช้กับตัวดำเนินการทางพันธุกรรมต่อไป [2]

3.2 การคัดเลือกแบบวงล้อรูเล็ตต์ (Roulette Wheel Selection)

การคัดเลือกแบบวงล้อรูเล็ตต์เป็นวิธีการคัดเลือกตามสัดส่วนของค่าความเหมาะสม การทำงานคล้ายกับการหมุนวงล้อรูเล็ตต์ [4] เมื่อวงล้อหยุดหมุน ส่วนที่ pointer ชี้คือค่าที่ถูกเลือก ดังรูปที่ 2 (ก) ขั้นตอนการทำงานของการคัดเลือกแบบวงล้อรูเล็ตต์มีดังนี้

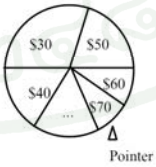
(1) คำนวณผลรวมค่าความเหมาะสมของประชากรทั้งหมด

(2) นำค่าความเหมาะสมของโครโมโซมแต่ละตัวมาหารด้วยค่าความเหมาะสมรวมที่คำนวณได้จากข้อ 1 ผลลัพธ์ที่ได้คือโอกาสของโครโมโซมที่จะถูกเลือกในวงล้อรูเล็ตต์ ดังสมการต่อไปนี้

$$P(a_i) = \frac{f(a_i)}{\sum_{j=1}^{n} f(a_j)}, \quad j = 1,2,\ldots,n \qquad (2)$$

เมื่อ n คือ จำนวนประชากรทั้งหมด
$f(a_i)$ คือ ค่าความเหมาะสมของโครโมโซม a_i
$P(a_i)$ คือ ความน่าจะเป็นที่โครโมโซม a_i จะถูกเลือก

(3) หมุนวงล้อ เมื่อวงล้อหยุดหมุนส่วนที่ Pointer ชี้คือโครโมโซมที่ถูกเลือก รหัสเทียมของการคัดเลือกแบบวงล้อรูเล็ตต์แสดงดังรูปที่ 2 (ข)




```
Algorithm : Roulette Wheel Selection()
    r := random number ,
                where 0 <= r < 1;
    sum := 0;
    for each individual i
    {   sum := sum + P(a_i);
        if( r < sum)
            return i;
    }
```

(ก) วงล้อรูเล็ตต์ (ข) รหัสเทียม

รูปที่ 2 การคัดเลือกแบบวงล้อรูเล็ตต์

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

3.3 การคัดเลือกแบบจัดลำดับ (Rank Selection)

การคัดเลือกแบบจัดลำดับเป็นการนำเอาโครโมโซมทั้งหมดมาเรียงลำดับตามค่าความเหมาะสม แล้วกำหนดค่าความเหมาะสมใหม่ โดยโครโมโซมที่มีค่าความเหมาะสมน้อยที่สุดจะได้ค่าความเหมาะสมเป็น 1 โครโมโซมที่มีค่าความเหมาะสมน้อยที่สุดเป็นลำดับที่ 2 จะได้ค่าความเหมาะสมเป็น 2 ส่วนโครโมโซมที่มีค่าความเหมาะสมสูงสุดจะได้ค่าความเหมาะสมเป็น n (เมื่อ n คือ จำนวนประชากรทั้งหมด) ตัวอย่างการกำหนดค่าความเหมาะสมใหม่แสดงดังตารางที่ 1

ตารางที่ 1 ค่าความเหมาะสมใหม่ของการคัดเลือกแบบจัดลำดับ

โครโมโซม	ค่าความเหมาะสมเดิม	ค่าความเหมาะสมใหม่
a_1	18	1
a_2	22	2
a_3	24	3
⋮	⋮	⋮
a_n	28	N

ค่าความเหมาะสมใหม่นี้จะนำไปผ่านขั้นตอนเช่นเดียวกับวงล้อรูเล็ตต์ได้โครโมโซมที่ถูกเลือก วิธีการคัดเลือกแบบจัดลำดับนี้ช่วยให้โครโมโซมที่มีค่าความเหมาะสมต่ำมากมีโอกาสถูกเลือกมากขึ้น [6]

4. การทดลอง

4.1 ปัญหาที่ใช้ในการทดลอง

ปัญหาที่ใช้ในการทดลองครั้งนี้มี 4 ปัญหา คือ OneMax, RandomMax, RoyalRoad, และ RandomRoyalRoad

4.1.1 OneMax

ปัญหา OneMax คือ ปัญหาการนับจำนวนเลขหนึ่งในโครโมโซม ค่าความเหมาะสมที่ได้ก็คือ จำนวนเลขหนึ่งทั้งหมดที่พบในโครโมโซม ดังนั้นโครโมโซมที่ดีที่สุดคือโครโมโซมที่ทุกบิตมีค่าเป็นหนึ่ง ปัญหา OneMax จำลองปัญหาการหาค่าที่เหมาะสมที่สุด (optimization problem) ที่ตัวแปรทุกตัวเป็นอิสระต่อกัน

4.1.2 RandomMax

ปัญหา RandomMax [7] คล้ายกับ OneMax เพราะเป็นปัญหาการนับจำนวนบิตที่ตรงกับคำตอบ แต่ต่างกันตรงที่คำตอบเกิดจากการสุ่ม โดยค่าความเหมาะสมที่ได้คือ จำนวนบิต ณ ตำแหน่งเดียวกับคำตอบที่มีค่าเท่ากัน เช่น

คำตอบที่สุ่มมา : 10100101

โครโมโซม : 10000100

ค่าความเหมาะสมที่ได้จะเป็นจำนวนบิตที่มีค่าตรงกับคำตอบซึ่งมีค่าเท่ากับ 6 เนื่องจากในบิตที่ 3 และ 8 (จากซ้าย) มีค่าไม่ตรงกับคำตอบที่สุ่มมา

4.1.3 RoyalRoad

ปัญหา RoyalRoad เป็นปัญหาที่จำลองความสัมพันธ์ของบิตในโครโมโซม โครโมโซมถูกแบ่งเป็นบล็อก คำตอบที่ต้องการคือทุกบิตในบล็อกมีค่าเท่ากับ 1 หากบิตใดบิตหนึ่งในบล็อกมีค่าไม่เท่ากับ 1 บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 แต่หากทุกบิตในบล็อกเป็น 1 หมด ค่าความเหมาะสมจะเท่ากับจำนวนบิตในบล็อกนั้น

4.1.4 RandomRoyalRoad

ปัญหา RandomRoyalRoad [7] คล้ายกับ RoyalRoad แต่ต่างกันตรงที่คำตอบของแต่ละบล็อกเกิดจากการสุ่ม ค่าของบิตในบล็อกที่เป็นคำตอบจึงมีได้ทั้ง 0 และ 1 ถ้าบิตในบล็อกของโครโมโซมที่ต้องการวัดค่าความเหมาะสมมีค่าเหมือนกับคำตอบ ค่าความเหมาะสมที่ได้จะเท่ากับจำนวนบิตในบล็อกนั้น แต่ถ้ามีเพียงบิตเดียวที่ค่าไม่เหมือนกับคำตอบ บล็อกนั้นจะได้ค่าความเหมาะสมเป็น 0 ทันที เช่น

คำตอบ : 1101 1101 1101 1101

โครโมโซม : 1101 1110 1101 1101

เนื่องจากในบล็อกที่ 1, 3, และ 4 มีค่าตรงกับคำตอบ แต่ในบล็อกที่ 2 มีบางบิตมีค่าไม่ตรงกับคำตอบ ดังนั้นค่าความเหมาะสมคือ 12

4.2 วิธีการทดลอง

4.2.1 วิธีการ

การทดลองนี้เปรียบเทียบประสิทธิภาพของวิธีการคัดเลือก 3 วิธีคือ การคัดเลือกแบบประลอง, แบบวงล้อรูเล็ตต์, และแบบจัดลำดับ โดยการคัดเลือกแต่ละวิธีจะใช้กับการไขว้เปลี่ยน 3 แบบ ได้แก่ การไขว้เปลี่ยนแบบจุดเดียว, แบบสองจุด, และแบบยูนิฟอร์ม แต่ละการทดลองทำซ้ำ 10 ครั้ง เปรียบเทียบประสิทธิภาพโดยใช้จำนวนรุ่นที่พบคำตอบหรือในกรณีที่ไม่พบคำตอบจะเปรียบเทียบค่าความเหมาะสมของโครโมโซม

4.2.2 พารามิเตอร์ที่ใช้ในการทดลอง

ตัวดำเนินการทางพันธุกรรมที่ใช้ในการทดลองนี้คือ การไขว้เปลี่ยน และการ shift and check โดยจะไม่มีการกลายพันธุ์ พารามิเตอร์ที่ใช้ในการทดลองแสดงในตารางที่ 2

ตารางที่ 2 พารามิเตอร์ที่ใช้ในการทดลอง

พารามิเตอร์	OneMax และ RoyalRoad	RandomMax และ RandomRoyalRoad
จำนวนประชากร	1,000	1,000
ความยาวโครโมโซม LZW	50	100
ค่าความเหมาะสมของคำตอบ	1,128	1,128
อัตราการ shift and check	50%	50%
อัตราการไขว้เปลี่ยน	80%	80%
อัตราการกลายพันธุ์	0%	0%
จำนวนรุ่นจำกัดไว้ที่	100	100 และ 200
ขนาดของการประลอง	4	4

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

5. ผลการทดลอง

ผลการทดลองกับปัญหา OneMax แสดงในตารางที่ 4 การคัดเลือกแบบประลองทำให้ LZWGA สามารถหาคำตอบพบโดยใช้จำนวนรุ่นน้อยกว่าการคัดเลือกแบบวงล้อรูเล็ตต์และแบบจัดลำดับเสมอ โดยผลการทดลองในปัญหา OneMax สำหรับการไขว้เปลี่ยนแบบต่างๆ เป็นดังนี้

- เมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม การคัดเลือกแบบประลองช่วยให้พบคำตอบได้เร็วกว่าแบบวงล้อรูเล็ตต์และแบบจัดลำดับ 3.59 เท่า และ 2.22 เท่าตามลำดับ
- ในการไขว้เปลี่ยนแบบจุดเดียว การคัดเลือกแบบประลองช่วยให้พบคำตอบได้เร็วกว่าแบบวงล้อรูเล็ตต์และแบบจัดลำดับ 1.46 เท่า และ 1.14 เท่าตามลำดับ
- ในการไขว้เปลี่ยนแบบสองจุด การคัดเลือกแบบประลองช่วยให้พบคำตอบได้เร็วกว่าแบบวงล้อรูเล็ตต์และแบบจัดลำดับ 1.53 เท่า และ 1.05 เท่าตามลำดับ

รูปที่ 3 แสดงค่าความเหมาะสมของโครโมโซมในขณะที่ LZWGA กำลังแก้ปัญหา OneMax จะเห็นว่าในระหว่างที่วิวัฒนาการกำลังดำเนินไป การคัดเลือกแบบประลองทำให้ได้โครโมโซมที่มีค่าความเหมาะสมมากกว่าการคัดเลือกแบบอื่น

ผลการทดลองกับปัญหา RoyalRoad แสดงในตารางที่ 5 การคัดเลือกแบบประลองทำให้ LZWGA สามารถหาคำตอบพบโดยใช้จำนวนรุ่นน้อยกว่าการคัดเลือกแบบวงล้อรูเล็ตต์และแบบจัดลำดับเสมอเช่นกัน โดยผลการทดลองในปัญหา RoyalRoad สำหรับการไขว้เปลี่ยนแบบต่างๆ เป็นดังนี้

- เมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม การคัดเลือกแบบประลองช่วยให้พบคำตอบได้เร็วกว่าแบบวงล้อรูเล็ตต์และแบบจัดลำดับ 3.57 เท่า และ 2.11 เท่าตามลำดับ
- ในการไขว้เปลี่ยนแบบจุดเดียว การคัดเลือกแบบประลองช่วยให้พบคำตอบได้เร็วกว่าแบบวงล้อรูเล็ตต์และแบบจัดลำดับ 1.75 เท่า และ 1.42 เท่าตามลำดับ
- ในการไขว้เปลี่ยนแบบสองจุดการคัดเลือกแบบประลองช่วยให้พบคำตอบได้เร็วกว่าแบบวงล้อรูเล็ตต์และแบบจัดลำดับ 1.57 เท่า และ 1.43 เท่าตามลำดับ

รูปที่ 4 แสดงค่าความเหมาะสมของโครโมโซมในขณะที่ LZWGA กำลังแก้ปัญหา RoyalRoad จะเห็นว่าในขณะที่วิวัฒนาการกำลังดำเนินไป การคัดเลือกแบบประลองทำให้ได้โครโมโซมที่มีค่าความเหมาะสมมากกว่าการคัดเลือกแบบอื่นเช่นเดียวกับในปัญหา OneMax

ผลการทดลองกับปัญหา RandomMax อยู่ในตารางที่ 6 และรูปที่ 5 ซึ่ง LZWGA หาคำตอบไม่พบและโครโมโซมที่ได้จากวิธีการคัดเลือกทั้ง 3 วิธี มีค่าความเหมาะสมไม่แตกต่างกันมากนัก แต่การคัดเลือกแบบประลองจะให้โครโมโซมที่มีค่าความเหมาะสมดีกว่าแบบวงล้อรูเล็ตต์ และแบบจัดลำดับเสมอถึงแม้จะดีกว่าไม่มากก็ตาม

ผลการทดลองกับปัญหา RandomRoyalRoad แสดงในตารางที่ 7 และรูปที่ 6 LZWGA หาคำตอบไม่พบเช่นเดียวกับปัญหา RandomMax แต่การคัดเลือกแบบประลองจะช่วยให้ได้โครโมโซมที่มีค่าความเหมาะสมมากที่สุดเมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม ถ้าใช้การไขว้เปลี่ยนแบบจุดเดียวและแบบสองจุด การคัดเลือกแบบวงล้อรูเล็ตต์จะทำให้โครโมโซมที่ได้มีค่าความเหมาะสมสูงสุด

ตารางที่ 4 จำนวนรุ่นเฉลี่ยที่พบคำตอบในปัญหา OneMax

	Tournament	Roulette Wheel	Rank
Uniform	27	97	60
One-Point	28	41	32
Two-Point	36	55	38

ตารางที่ 5 จำนวนรุ่นเฉลี่ยที่พบคำตอบในปัญหา RoyalRoad

	Tournament	Roulette Wheel	Rank
Uniform	28	100	59
One-Point	24	42	34
Two-Point	28	44	40

ตารางที่ 6 ค่าความเหมาะสมเฉลี่ยในปัญหา RandomMax

	Tournament	Roulette Wheel	Rank
Uniform	623.7	589.1	596.3
One-Point	616.4	600.8	608.0
Two-Point	608.0	595.2	598.0

ตารางที่ 7 ค่าความเหมาะสมเฉลี่ยในปัญหา RandomRoyalRoad

	Tournament	Roulette Wheel	Rank
Uniform	676.0	358.0	395.2
One-Point	418.0	619.6	594.0
Two-Point	381.2	636.4	561.2

6. สรุป

ในปัญหา OneMax, RoyalRoad, และ RandomMax วิธีการคัดเลือกที่ทำให้ LZWGA ทำงานได้อย่างมีประสิทธิภาพมากที่สุดคือการคัดเลือกแบบประลอง ไม่ว่าจะใช้ร่วมกับการไขว้เปลี่ยนแบบใด ส่วนในปัญหา RandomRoyalRoad การคัดเลือกแบบประลองจะมีประสิทธิภาพดีเมื่อใช้ร่วมกับการไขว้เปลี่ยนแบบยูนิฟอร์ม แต่เมื่อใช้การไขว้เปลี่ยนแบบจุดเดียวและสองจุด วิธีการคัดเลือกที่ดีที่สุดคือการคัดเลือกแบบวงล้อรูเล็ตต์

ผลการทดลองแสดงว่าวิธีการคัดเลือกมีผลต่อประสิทธิภาพการทำงานของ LZWGA โดยการคัดเลือกแบบประลองทำให้ LZWGA มีประสิทธิภาพในการทำงานมากที่สุดเช่นเดียวกับในงานวิจัย [4] ซึ่งรายงานว่าวิธีการคัดเลือกที่ดีที่สุดสำหรับ GA คือการคัดเลือกแบบประลอง และเนื่องจากวิธีการคัดเลือกมีผลต่อประสิทธิภาพการทำงานของ LZWGA ผู้วิจัยจึงมีความคิดที่จะพัฒนาวิธีการคัดเลือกเพื่อเพิ่มประสิทธิภาพการทำงานของ LZWGA โดยรวมการคัดเลือกแบบประลองเข้ากับการคัดเลือกแบบอื่น ซึ่งจะได้ศึกษาต่อไป

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

เอกสารอ้างอิง

[1] Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Professional, USA, 1989.

[2] Handcock, P.J.B, "An Empirical Comparison of Selection Methods in Evolutionary Algorithm," Proceedings of the AISB Workshop on Evolutionary Computation, 1994.

[3] นริศ กุณาศล, วรเศรษฐ สุวรรณิก, และ ประภาส จงสถิตย์วัฒนา, "การเข้ารหัสแบบ LZW ในขั้นตอนวิธีเชิงพันธุกรรม," Proceedings of Electrical Engineering Conference (EECON-28), หน้า 861-864, 20-21 ตุลาคม, 2548.

[4] Zhong J., Hu X., Gu M., and Zhang J., "Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms," Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05), 2005.

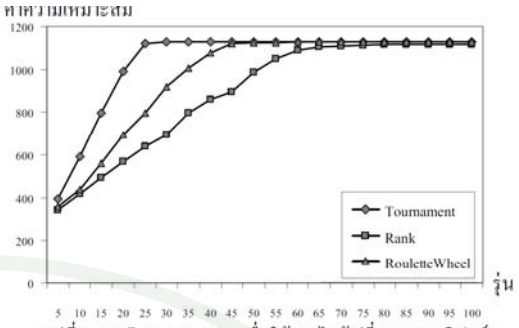
[5] อณิมา รอดเสียงล้ำ และ วรเศรษฐ สุวรรณิก, "ตัวดำเนินการ Shift สำหรับ LZWGA," การประชุมทางวิชาการระดับชาติทางด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 5, The 5th National Conference on Computing and Information Technology (NCCIT), 22-23 พฤษภาคม, 2552.

[6] Mitchell M., "An Introduction to Genetic Algorithms", The MIT Press, England, 1996.

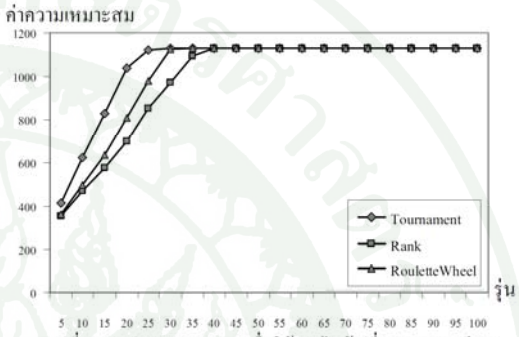
[7] สาวิตรี ฉ่ำลบัย และ วรเศรษฐ สุวรรณิก, "ปัญหาสำหรับทดสอบขั้นตอนวิธีเชิงพันธุกรรมแบบบีบอัด," Proceedings of Electrical Engineering Conference (EECON-30), หน้า 653-656, 25-26 ตุลาคม, 2550.

นางสาวอณิมา รอดเสียงล้ำ
การศึกษา : วิทยาศาสตรบัณฑิต มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา
ปัจจุบัน : ศึกษาระดับปริญญาโท สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
งานวิจัยที่สนใจ : ขั้นตอนวิธีเชิงพันธุกรรม

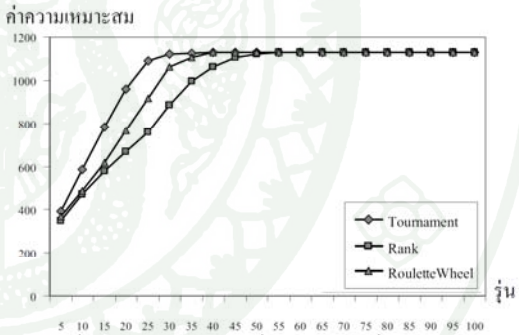
ดร.วรเศรษฐ สุวรรณิก
การศึกษา : วิศวกรรมศาสตรดุษฎีบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย
ปัจจุบัน : อาจารย์ประจำ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
งานวิจัยที่สนใจ : ขั้นตอนวิธีเชิงพันธุกรรม



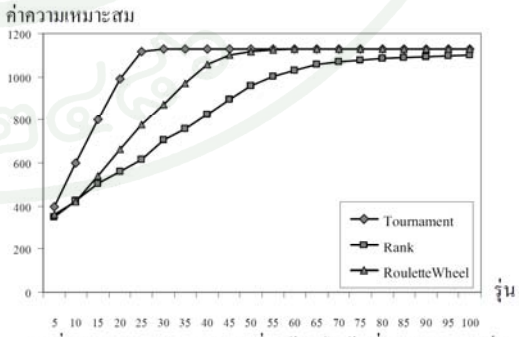
รูปที่ 3 (ก) ปัญหา OneMax เมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม



รูปที่ 3 (ข) ปัญหา OneMax เมื่อใช้การไขว้เปลี่ยนแบบจุดเดียว

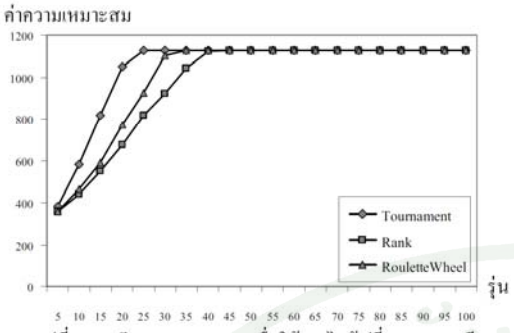


รูปที่ 3 (ค) ปัญหา OneMax เมื่อใช้การไขว้เปลี่ยนแบบสองจุด

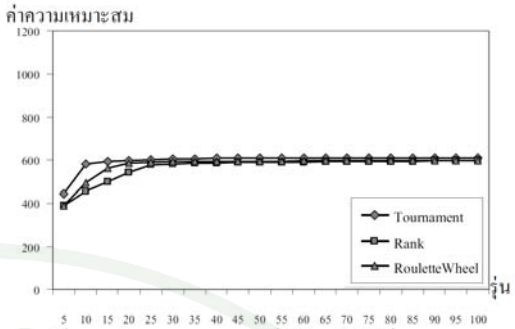


รูปที่ 4 (ก) ปัญหา RoyalRoad เมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม

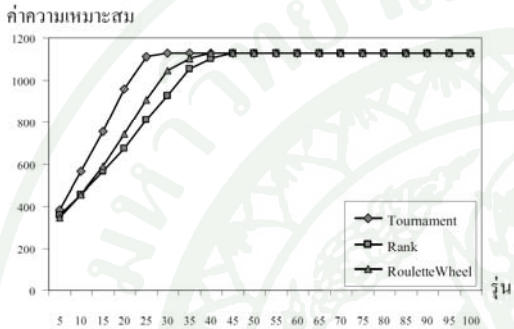
ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์



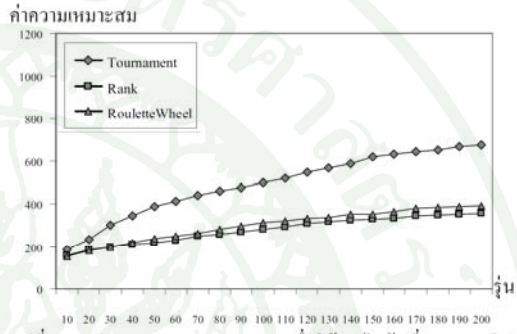
รูปที่ 4 (ข) ปัญหา RoyalRoad เมื่อใช้การไขว้เปลี่ยนแบบจุดเดียว



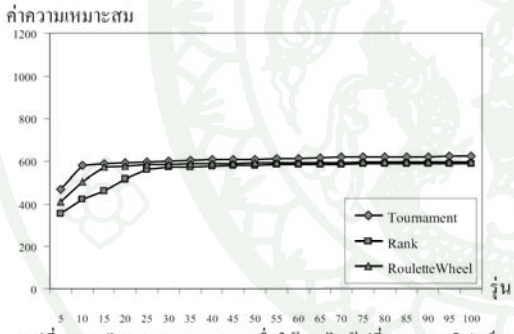
รูปที่ 5 (ค) ปัญหา RandomMax เมื่อใช้การไขว้เปลี่ยนแบบสองจุด



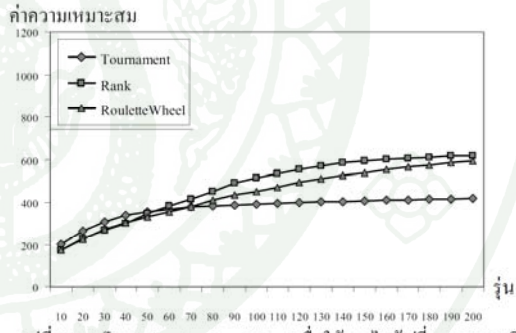
รูปที่ 4 (ค) ปัญหา RoyalRoad เมื่อใช้การไขว้เปลี่ยนแบบสองจุด



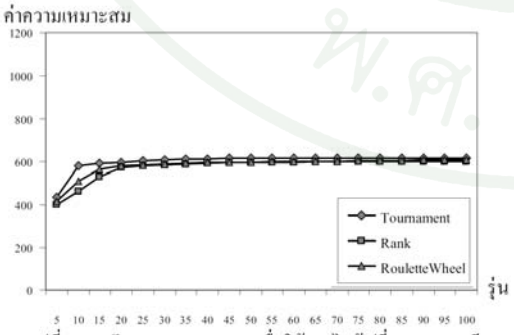
รูปที่ 6 (ก) ปัญหา RandomRoyalRoad เมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม



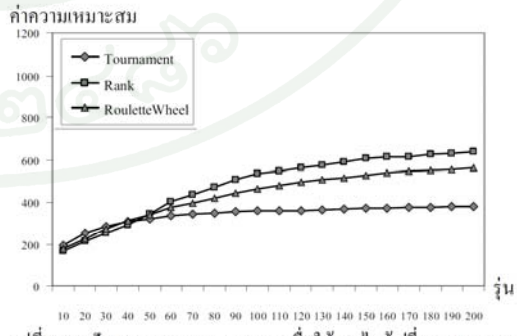
รูปที่ 5 (ก) ปัญหา RandomMax เมื่อใช้การไขว้เปลี่ยนแบบยูนิฟอร์ม



รูปที่ 6 (ข) ปัญหา RandomRoyalRoad เมื่อใช้การไขว้เปลี่ยนแบบจุดเดียว



รูปที่ 5 (ข) ปัญหา RandomMax เมื่อใช้การไขว้เปลี่ยนแบบจุดเดียว



รูปที่ 6 (ค) ปัญหา RandomRoyalRoad เมื่อใช้การไขว้เปลี่ยนแบบสองจุด

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

An Improved Shift-and-Check Operator for LZWGA

Animar Rotsianglum and Worasait Suwannik
Department of Computer Science, Kasetsart University, Bangkok, Thailand.
Email: animar_tookta@hotmail.com, worasait.suwannik@gmail.com

Abstract

LZW encoding in Genetic Algorithm (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. This encoding reduces the length of the chromosome and enables the algorithm to solve very large problems. In our previous work, a shift-and-check operator was proposed to improve the performance of LZWGA. The operator randomly selects a shift position and left-shifts integers in an LZWGA chromosome. This paper presents an improved shift-and-check operator which the shifting probability is different for each position. The result shows that the new operator can improve the LZWGA performance.

Keywords: Genetic Algorithm, LZWGA, Operator, Shift.

1. Introduction

Genetic algorithm (GA) is a search technique that imitates the natural evolution process [1]. To solve a problem using GA, solutions are encoded in binary strings chromosome. Larger chromosome means the larger search space. For instance, the search space size is 2^{128} for 128-bits OneMax problem. But for one million bits OneMax problem, the search space size will be $2^{1000000}$.

To reduce the search space size, LZWGA was proposed [2]. LZWGA is developed from GA. LZWGA uses a compressed encoding. The length of a compressed chromosome is shorter than that of an uncompressed chromosome. A compressed chromosome is decompressed before its fitness can be evaluated. For example, a compressed chromosome {1,0,2,4} can be decompressed to a binary string 1010101.

The LZWGA chromosome is an integer array. Shifting integers in a chromosome for one position often affect its fitness value. In our previous work [3], we proposed an operator called "shift-and-check". This operator shifts a position in LZWGA chromosome. The result shows that LZWGA with the shift-and-check operator has better performance than the original LZWGA.

Using the shift-and-check operator in different positions in a chromosome results in various fitness values. We hypothesized that the probability of this operator for each position should not be equal. The probability should be different according to a position in a chromosome. We conducted a preliminary experiment to test our hypothesis as explained in section 3. From the result, we propose an improved shift-and-check operator, which adjusts shifting probabilities for different positions in a chromosome. The improved shift-and-check operator is shown in section 4.

2. LZWGA

2.1 Algorithm

The main difference between LZWGA and simple GA is that LZWGA chromosome is in a compressed format. Therefore, it has to be decompressed before its fitness can be evaluated. The flowchart of LZWGA is shown in figure 1.

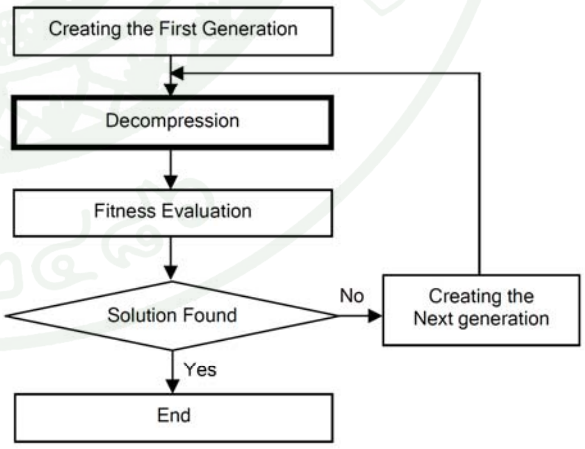


Figure 1. LZWGA Flowchart.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

The algorithm begins by creating the first generation of compressed chromosomes. Before evaluating the fitness of each chromosome, the compressed chromosome is decompressed using the LZW decompression algorithm. After that, the new population is created and replaces the old population. The algorithm repeats the process of decompression, fitness evaluation, and creating a new population until the termination criterion is met. The algorithm terminates when a solution is found or a maximum generation is reached.

2.2 Chromosome

Unlike a simple GA, the LZWGA chromosome is integer encoding. A chromosome in LZWGA is in a compressed format. Each integer in a chromosome is a code for an index of an entry in a compression dictionary. Chromosomes in the first generation are created randomly with the constraint that the i^{th} integer of a chromosome must not have value greater than i+1. Otherwise, the dictionary look up in will be failed after the $(i+1)^{th}$ digit is read.

LZWGA chromosomes in the first generation are randomly generated from equation 1.

$$a_i = r \bmod (i+1),\ i \in \{1, 2, 3, \ldots, n\} \qquad (1)$$

Where a_i is a value of the i^{th} position in LZWGA chromosome
r is a random integer
n is the length of an LZWGA chromosome

2.3 Decompression

Because the chromosome in LZWGA is compressed, it has to be decompressed before its evaluation. A compressed chromosome is decompressed using LZW decompression algorithm [4].

The length of the output chromosome is not fixed. If the length is more than the size of the solution, the excess bits are discarded. If the length is less than the size of the solution, the remaining bits are filled with 0's or the fitness evaluation would be stopped when reading to the end of the string. Then, the fitness of the decompressed binary string is evaluated. The fitness of a compressed chromosome is equals to the fitness of a decompressed chromosome.

2.4 Operators for LZWGA

Similar to GA, LZWGA uses the current population to create the next generation. The two genetic operators for LZWGA are crossover and mutation.

- Crossover is a genetic operator that combines two parent chromosomes to produce new offsprings [5].
- Mutation is a genetic operator that positions in a chromosome are randomly mutated.

3. Background

3.1 Shift-and-Check Operator (SC)

In our previous work, we presented a new operator for LZWGA called "shift-and-check" (SC). The SC operator randomly left-shifts numbers in an LZWGA chromosome. The SC algorithm was explained in [3].

After shifting, we prevent the value of the i^{th} position of chromosome from becoming 0 (except when the position is already 0). That is if $a_{i+1} \bmod (i+1) = 0$ and a_{i+1} is not equal to 0 then the value of the i^{th} position is obtained from equation 2.

$$a_i = a_{i+1} - 1 \qquad (2)$$

3.2 Background of the proposed operator

A new mutation method for LZWGA was suggested in [6]. Different mutation rates that depend on positions were proposed. From this research, we got an idea to observe our SC operator in every position. We would like to know about the change in fitness values when SC at different positions.

To observe the change, we generated an original chromosome. Then we used an SC operator in every position and evaluated the fitness using OneMax function. The experiment was repeated 1,000 times. The observation was started at 4^{th} to 15^{th} positions of LZWGA chromosome. Average fitness values were shown in table 1.

From table 1, a fitness value was varied depend on a shifted position. Shifting at left positions often produced higher fitness value than right positions. From this result, probability for using an SC operator should not be equal in every position. It should be different according to the position in LZWGA chromosome.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

Table 1. Average fitness values from using SC operator at every position of LZWGA chromosome.

Chromosome length	Shifted Position														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	5.2	3.7	3.1	2.7											
5	7.3	5.2	4.5	4.0	3.7										
6	9.8	6.9	6.0	5.4	5.0	4.8									
7	12.2	9.1	8.1	7.1	6.6	6.2	6.0								
8	14.9	11.1	9.4	8.5	8.0	7.5	7.3	7.1							
9	17.1	13.0	11.3	10.3	9.6	9.2	8.8	8.5	8.3						
10	19.4	14.7	13.1	12.0	11.1	10.6	10.3	10.0	9.7	9.6					
11	22.2	16.6	15.0	13.4	12.5	12.0	11.6	11.4	11.0	10.8	10.7				
12	24.8	19.2	16.8	15.7	14.7	14.0	13.6	13.2	12.8	12.6	12.5	12.3			
13	28.8	21.9	19.2	17.3	16.4	15.6	15.2	14.7	14.4	14.1	13.8	13.6	13.5		
14	30.9	23.4	20.8	19.2	18.4	17.4	17.0	16.4	16.1	15.8	15.6	15.4	15.2	15.0	
15	34.6	26.8	23.9	21.7	20.7	19.9	19.3	18.5	18.3	18.0	17.7	17.5	17.3	17.1	17.0

4. Improved Shift-and-Check Operator (Improved SC)

This section explains our method for improving the SC operator. We produce different probabilities for using this operator by assigning weigh for every position of chromosome.

P_i denotes probability that the SC operator is used at the i^{th} position. It is defined as follows:

$$P_i = \frac{w_i}{\sum_{j=1}^{n} w_j}, j=1,2,\ldots,n \quad (3)$$

Where i is the current position in chromosome
j is the position in chromosome
n is the length of chromosome
w_i is the chromosome weight of the i^{th} position, the value of w_i is defined as follows:

$$w_i = (n+1)-i \quad (4)$$

For instance, we have a 5-positions LZWGA chromosome and values of w_i and P_i were shown in table 2. The first position has the most chance to be chosen and chances of the next positions will decrease slowly.

Table 2.Values of w_i and P_i of each position in a 5-positions LZWGA chromosome.

Position	1	2	3	4	5
w_i	5	4	3	2	1
P_i	0.33	0.27	0.20	0.13	0.07

5. Experiment

5.1 Test Problems

Our test problems are OneMax, RoyalRoad, and RandomRoyalRoad.

5.1.1 OneMax

OneMax is a simple problem. The objective of this problem is to maximize the number of 1's in a bit string. This problem models an optimization problem that every variable is independent.

For example,
Chromosome : 1111101111011111

A fitness value of this chromosome is 14 because this chromosome contains 14 bits of 1's.

5.1.2 RoyalRoad

RoyalRoad problem models bit relationship within a chromosome. A chromosome is divided into n blocks. Each block is k bits long. The solution is all block contains 1's. If every bit in block are 1's, the fitness of that block is equal to block size, but if a block contains at least one 0. Its fitness is zero.

For example,
Chromosome : 1111 1011 1101 1111

Every bit in the 1st and 4th block are 1's. Therefore the fitness value of this chromosome is 8.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

5.1.3 RandomRoyalRoad

RandomRoyalRoad problem resembles RoyalRoad but the difference is a solution block is generated randomly [7]. If all bits in a block are the same as the bits in the solution block, the fitness of that block is equal to the block size. But if a block differs from the solution block, a fitness of that block is zero.

For example,

Solution : 1101 1101 1101 1101
Chromosome : 1101 1110 1101 1101

The 2nd block differs from the solution block. The 1st, 3rd, and 4th are similar to the solution block, so the fitness value of this chromosome is 12 (3 equal blocks × 4 bits per block).

5.2 Parameters

We used the same parameters (shown in table 3 and 4) for 3 methods, which are original LZWGA, LZWGA with SC operator, and LZWGA with improved SC operator. Each experiment was repeated for 10 times. We compared the performance by using fitness values.

Experimental parameters are shown in table 3 and 4. We focused on large problems, small population size, and no mutation. For crossover and selection methods, we used uniform crossover and tournament selection because they work well for LZWGA [8, 9].

Table 3.
Parameter for OneMax and RoyalRoad problems.

Parameter	Value
Population size	300
LZWGA Chromosome length	1,000
Solution fitness	100,000
Selection method	Tournament selection Tournament size=4
Crossover method	Uniform crossover
Crossover rate	80%
Shift rate	50%
Max Generation	1,000
Block size	10

Table 4. Parameter for RandomRoyalRoad problem.

Parameter	Value
Population size	300
LZWGA Chromosome length	2,000
Solution fitness	100,000
Selection method	Tournament selection Tournament size=4
Crossover method	Uniform crossover
Crossover rate	80%
Shift rate	50%
Max Generation	1,000
Block size	4

6. Result

For OneMax and RoyalRoad problems, both SC and improved SC can help LZWGA found a solution. Tabel 5 shows the generation that a solution can be found. LZWGA with improved SC found a solution faster than LZWGA with SC 269 and 247 generations in OneMax and RoyalRoad problems respectively, but original LZWGA cannot find a solution.

Table 5. The generation that a solution can be found.

Problems	OneMax	RoyalRoad
Original LZWGA	Not found	Not found
LZWGA with SC	951	998
LZWGA with improved SC	682	751

For RandomRoyalRoad problem, a solution cannot be found by using LZWGA. The best fitness value from 3 methods was shown in table 6. Chromosomes from LZWGA with improved SC have better fitness value than LZWGA with SC 1.09 times and better than original LZWGA 1.70 times. The improved SC operator helped LZWGA produced the best chromosome.

Table 6. The best fitness value in RandomRoyalRoad

Problems	RandomRoyalRoad
Original LZWGA	6256
LZWGA with SC	9778
LZWGA with improved SC	10616

Figure 2 shows best fitness values from original LZWGA, LZWGA with SC, and LZWGA with improved SC when solving OneMax, RoyaRoad, and RandomRoyalRoad.

The figure shows that original LZWGA produced the worst chromosomes. In addition, the improved SC operator produced chromosomes that have better fitness value than original LZWGA and LZWGA with SC in every generation of our test problems. Thus, the improved SC operator can enhance the performance of LZWGA.

7. Conclusion

This paper proposed an improved shift-and-check operator or improved SC. The operator assigns different probabilities for using the shift-and-check operator for each position of LZWGA chromosome. The improved SC operator can improve the performance of LZWGA because the operator helped LZWGA to find a solution faster than the SC operator 1.39 times in OneMax problem and 1.32 times in RoyalRoad problem. For RandomRoyalRoad problem, a solution cannot be found. But LZWGA with improved SC produced better chromosome than

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

SC 1.09 times and produced better chromosome than original LZWGA 1.70 times.

Although LZWGA with the improved SC operator has good efficiency, but for difficult problems (e.g., RandomRoyalRoad), solutions still cannot be found. In the future, we will keep on finding a new technique to solve this problem.

8. References

[1] D.E.Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, USA, 1989.

[2] N. Kunasol, W. Suwannik, and P. Chongstitvatana, "LZWEncoding in Genetic Algorithm," Proceedings of Electrical Engineering Conference (EECON-28), pages 861-864, October 20-21, 2005. (abstract in English)

[3] A. Rotsianglum, and W. Suwannik, "A Shift Operator for LZWGA," The 5th National Conference on Computing and Information Technology (NCCIT), page 320-325, May 22-23, 2009. (abstract in English)

[4] H. Yokoo, "Improved Variations Relating the Ziv-Lempel and Welch-Type Algorithm for Sequential Data Compression," IEEE Trans on Information Theory, Vol. 30, No. 1, page 73-81, 1992.

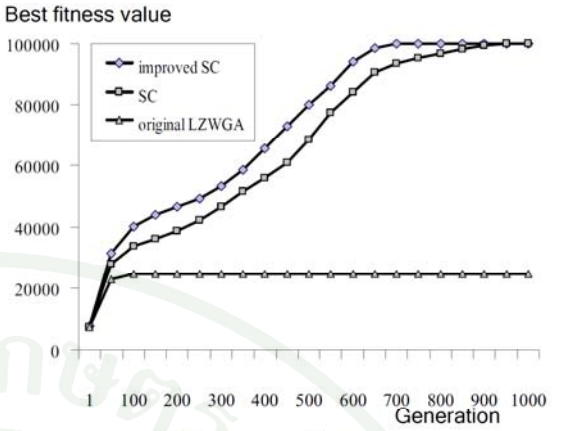
[5] W. Suwannik, N. Kunasol, and P. Chongstitvatana, "Compressed Genetic Algorithm," Proceedings of Northeastern Computer Science and Engineering Conference (NECSEC), page 203-211, March 31- April 1, 2005. (abstract in English)

[6] S. Numnark, and W. Suwannik, "Improving the Performance of LZWGA by Using a New Mutation Method," Proceedings of IEEE Congress on Evolutionary Computation (CEC 2008), page 1862-1865. June 1-6, 2008.

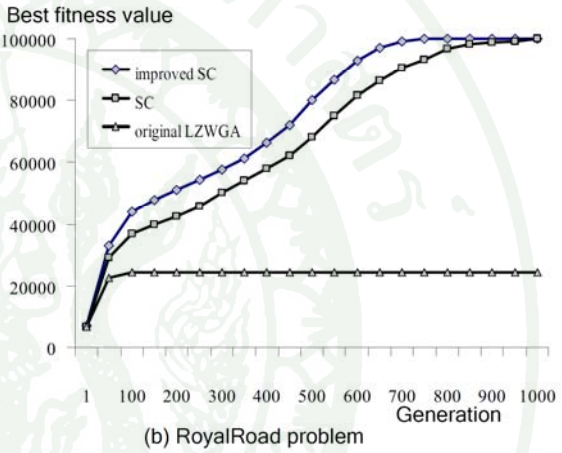
[7] S. Chumlamai, and W.Suwannik, "Benchmark Problems for Compressed Genetic Algorithm," Proceedings of Electrical Engineering Conference (EECON-30), page 653-656, October 25-26, 2007. (abstract in English)

[8] S. Numnark, and W. Suwannik, "A Comparison of Crossover Methods in LZWGA," The 3rd National Conference on Computing and Information Technology (NCCIT), page 328-333, May 25-26, 2007. (abstract in English)

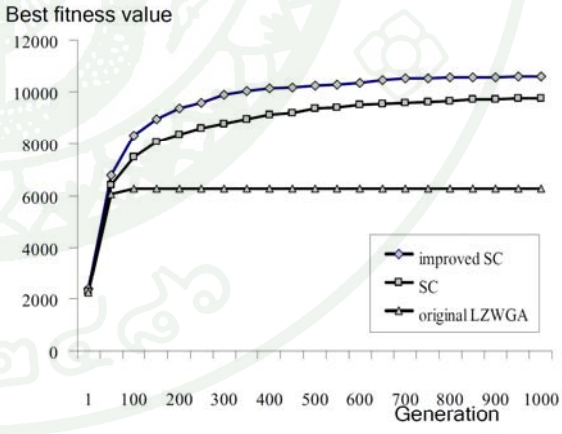
[9] A. Rotsianglum, and W. Suwannik, "Comparing the Performance of LZWGA with Various Selection Methods," National Computer Science and Engineering Conference 2009 (NCSEC 2009), page 198-203, November 4-6, 2009. (abstract in English)



(a) Onemax problem



(b) RoyalRoad problem



(c) RandomRoyalRoad problem

Figure 2. Fitness values from original LZWGA, LZWGA with SC, and LZWGA with improved SC.

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

ประวัติการศึกษา และการทำงาน

ชื่อ	นางสาวอณิมา รอดเสียงล้ำ
เกิดวันที่	28 มกราคม 2525
สถานที่เกิด	อำเภอวิเศษชัยชาญ จังหวัดอ่างทอง
ประวัติการศึกษา	วท.บ. (วิทยาการคอมพิวเตอร์) มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา (เกียรตินิยมอันดับสอง) ประกาศนียบัตรบัณฑิต วิชาชีพครู มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา
ตำแหน่งปัจจุบัน	ครู อันดับ คศ.1
สถานที่ทำงานปัจจุบัน	โรงเรียนหอวัง สังกัดสำนักงานเขตพื้นที่การศึกษากรุงเทพมหานคร เขต 2
ผลงานดีเด่นและ/หรือรางวัลทางวิชาการ	-
ทุนการศึกษาที่ได้รับ	ได้รับทุนโครงการส่งเสริมการผลิตครูที่มีความสามารถพิเศษทางวิทยาศาสตร์และคณิตศาสตร์(สควค.) จากสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท.)

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์