

ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

ศรีชล ภิรมย์ลาภ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
พฤษภาคม 2560
ลิขสิทธิ์เป็นของมหาวิทยาลัยบูรพา

A FREQUENCY-BASED UPDATING STRATEGY IN COMPACT GENETIC ALGORITHM

SRICHOL PHIROMLAP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE MASTER DEGREE OF SCIENCE
IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATICSBURAPHA UNIVERSITY
MAY 2017
COPYRIGHT OF BURAPHA UNIVERSITY

คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ได้พิจารณา
วิทยานิพนธ์ของ ศรีชด ภิรมย์ตาก ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

.....*ศิริ อิมอร*.....อาจารย์ที่ปรึกษาหลัก
(ผู้ช่วยศาสตราจารย์ ดร.สุนิสา ริมเจริญ)

คณะกรรมการสอบวิทยานิพนธ์

.....*เฉลิมทรัพย์ สังขวิจิตร*..... ประธานกรรมการ
(ดร.เฉลิมทรัพย์ สังขวิจิตร)

.....*ศิริ อิมอร*..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุนิสา ริมเจริญ)

.....*ณัฐนันท์ ลีลาตระกูล*..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ณัฐนันท์ ลีลาตระกูล)

คณะวิทยาการสารสนเทศ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยบูรพา

.....*ก. อิมอร*..... คณบดีคณะวิทยาการสารสนเทศ
(ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร)

วันที่ 31 เดือน พฤษภาคม พ.ศ. 2560

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาจาก ผศ.ดร.สุนิสา ริมเจริญ อาจารย์ที่ปรึกษาหลัก ที่ได้สอน และให้คำปรึกษาในการดำเนินงานวิจัย ช่วยวิเคราะห์ถึงการแก้ไขปัญหาต่างๆ ในการพัฒนาและปรับปรุงประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับที่ผู้วิจัยดำเนินงานวิจัยอยู่ และขอบคุณที่อดทน และเอาใจใส่ด้วยดีเสมอมา ขอบคุณที่อดทนอ่านผลงานวิจัย และช่วยแนะนำ แก้ไขให้งานวิจัยมีความถูกต้องสมบูรณ์ ผู้วิจัยรู้สึกซาบซึ้งเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ ผศ.ดร.ณัฐนนท์ ลีลาตระกูล ที่กรุณาให้คำปรึกษา และแนะนำการคำนวณประสิทธิภาพของงานวิจัย ทำให้งานวิจัยมีความสมบูรณ์มากยิ่งขึ้น นอกจากนี้ยังได้ให้คำปรึกษา และให้กำลังใจ ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

เนื่องจากผู้วิจัยได้รับทุนการศึกษาจากมหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา จึงขอขอบพระคุณ ณ ที่นี้ด้วย

ขอกราบขอบพระคุณ คุณแม่แจ่มซ้อย ศิริสรรพ ที่คอยเป็นที่ปรึกษา และเป็นกำลังใจ และชี้แนะแนวทางต่าง ๆ ในการดำเนินการศึกษาให้ประสบความสำเร็จ และเพื่อนนิสิตปริญญาโท คณะวิทยาการสารสนเทศ ทุกคนที่ให้คำแนะนำ กำลังใจและให้การสนับสนุนผู้วิจัยเสมอมา

ศรียล ภิรมย์ลาภ

55910168: สาขาวิชา: เทคโนโลยีสารสนเทศ; วท.ม. (เทคโนโลยีสารสนเทศ)

คำสำคัญ: ขั้นตอนวิธีเชิงพันธุกรรม / ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ / ปัญหาทับคัก

ศรีชล ภิรมย์ลาภ : ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ (A

FREQUENCY-BASED UPDATING STRATEGY IN COMPACT GENETIC ALGORITHM)

คณะกรรมการควบคุมวิทยานิพนธ์:สุนิสา ริมเจริญ, Ph.D. 62 หน้า. ปี พ.ศ. 2560.

ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ เป็นหนึ่งในขั้นตอนวิธีเชิงวิวัฒนาการแบบใหม่ ที่มีแนวความคิดในการใช้ตัวแบบความน่าจะเป็น (Probabilistic Model) แทนการใช้กลุ่มประชากรแบบเดิมในการค้นหาคำตอบ แนวความคิดนี้ทำให้ขั้นตอนวิธีเชิงพันธุกรรมใช้หน่วยความจำในการเก็บประชากรน้อยลง อีกทั้งยังไม่ต้องอาศัยการดำเนินการเชิงพันธุกรรม เช่น การไขว้เปลี่ยน หรือ การกลายพันธุ์ ทำให้การประมวลผลทำได้รวดเร็วยิ่งขึ้น โดยที่ยังคงความสามารถเทียบเท่ากับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายที่ใช้อยู่เดิม งานวิจัยนี้นำเสนอการปรับปรุงวิธีการปรับค่าเวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยเรียกขั้นตอนที่นำเสนอใหม่นี้ว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ โดยขั้นตอนวิธีจะจัดเก็บข้อมูลค่าความถี่ และค่าความต่อเนื่องในการปรับค่าความน่าจะเป็นเข้าหาค่า 0.0 หรือ 1.0 เพื่อนำมาใช้เป็นข้อมูลในการตัดสินใจ และกำหนดขนาดของการปรับค่าความน่าจะเป็นในแต่ละรอบการทำงาน จากผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอสามารถหาคำตอบได้ดีกว่า หรือใกล้เคียงกับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับแบบเดิม โดยใช้จำนวนครั้งในการประเมินค่าความเหมาะสมน้อยกว่า

55910168: MAJOR: INFORMATION TECHNOLOGY; M.Sc.

(INFORMATION TECHNOLOGY)

KEYWORDS: GENETIC ALGORITHM / COMPACT GENETIC ALGORITHM / TRAP
FUNCTION

SRICHOL PHIROMLAP: A FREQUENCY-BASED UPDATING STRATEGY IN
COMPACT GENETIC ALGORITHM. ADVISORY COMMITTEE: SUNISA RIMCHAROEN,
Ph.D. 62 P. 2017.

The Compact Genetic Algorithm (cGA) is one of evolutionary algorithms. It employs a probabilistic model instead of using population in searching for solutions. This makes the compact genetic algorithm take advantage in using a small amount of memory and eliminating genetic operators such as crossover and mutation. This technique helps the algorithm to process faster. It is comparable with Simple Genetic Algorithm (sGA). This thesis proposes an adaptation of updating strategy in the compact genetic algorithm to help the algorithm to achieve a higher solution quality with fewer evaluations. We named the proposed technique as the frequency based compact genetic algorithm (fb-cGA). The fb-cGA collects information from the past, i.e. frequencies and continuity of updating probabilities towards 0.0 and 1.0. The frequencies and continuity are used to guide an updating step size. The experiment results show that our proposed method requires lesser number of fitness evaluations and achieves similar or higher solution quality compared with the original cGA.

สารบัญ

	หน้า
กิตติกรรมประกาศ.....	ง
บทคัดย่อภาษาไทย.....	จ
บทคัดย่อภาษาอังกฤษ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่	
1 บทนำ	
ความเป็นมาและความสำคัญของปัญหา.....	1
วัตถุประสงค์ของการวิจัย.....	2
แนวทางในการศึกษา วิเคราะห์ และออกแบบอัลกอริทึม.....	2
ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย.....	2
ขอบเขตของการวิจัย.....	2
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	
ขั้นตอนวิธีเชิงพันธุกรรม.....	4
การทำงานของขั้นตอนวิธีเชิงพันธุกรรม.....	4
การเข้ารหัสโครโมโซม (Chromosome Encoding).....	5
ผู้สร้างประชากรต้นกำเนิด.....	6
คำนวณค่าความเหมาะสมของประชากรแต่ละตัว.....	6
ดำเนินการค้นหาคำตอบด้วยวิธีทางพันธุกรรม.....	7
การคัดเลือกสายพันธุ์.....	7
การไขว้เปลี่ยน (Crossover).....	7
การกลายพันธุ์ (Mutation).....	8
ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ.....	9
การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ.....	10
ปัญหาที่ใช้ทดสอบ.....	11
ปัญหาจำนวนบิตหนึ่งมากที่สุด.....	11

สารบัญ (ต่อ)

	หน้า
ปัญหาเรนคอมเม็ก.....	12
ปัญหารอยัลโรค.....	13
ปัญหาทับด้ก.....	15
งานวิจัยที่เกี่ยวข้อง.....	17
3 วิธีดำเนินงาน	
การศึกษาขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ.....	19
การศึกษาค่าความถี่ และค่าความต่อเนื่องในการปรับค่าของเวกเตอร์ความน่าจะเป็น.....	22
อัลกอริทึมที่นำเสนอ.....	24
การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่.....	27
การจัดเก็บข้อมูลค่าความถี่.....	27
การจัดเก็บข้อมูลค่าความต่อเนื่อง.....	28
การปรับปรุงค่าความน่าจะเป็น.....	29
ถ้าผู้ชนะมีค่าเป็น 1.....	30
ถ้าผู้ชนะมีค่าเป็น 0.....	30
พารามิเตอร์ p_{size}/n	31
4 ผลการทดลอง	
ขั้นตอนการทดลอง.....	35
พารามิเตอร์ที่ใช้ในการทดลอง.....	36
ผลการทดลอง.....	37
ผลการทดลองกับปัญหาจำนวนบิตหนึ่งมากที่สุด.....	37
ผลการทดลองกับปัญหาเรนคอมเม็ก.....	39
ผลการทดลองกับปัญหารอยัลโรค.....	40
ผลการทดลองกับปัญหาทับด้ก.....	43
การวิเคราะห์เวกเตอร์ความน่าจะเป็น.....	46
5 อภิปรายและสรุปผล	
ผลการวิจัย.....	50
ข้อเสนอแนะ.....	51

สารบัญ (ต่อ)

	หน้า
บรรณานุกรม.....	52
ภาคผนวก	
ภาคผนวก ก.....	55
ภาคผนวก ข.....	61
ประวัติย่อของผู้วิจัย.....	62

สารบัญตาราง

ตารางที่	หน้า
2-1 ตัวอย่างปัญหาจำนวนบิตหนึ่งมากที่สุด.....	12
2-2 ตัวอย่างปัญหาแรนคอมแม็ก.....	13
2-3 ตัวอย่างปัญหากับคัท.....	16
3-1 ค่าความถี่ และค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 และ 0.0.....	23
3-2 ตัวอย่างการจัดเก็บค่าความถี่ในการปรับปรุงค่าความน่าจะเป็น.....	28
3-3 ตัวอย่างการจัดเก็บค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็น.....	29
3-4 ผลการวัดอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมิน.....	32
3-5 ค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมิน โดยเฉลี่ยจากทุกปัญหา.....	33
และขนาดของ tournament	
4-1 พารามิเตอร์ที่ใช้กับปัญหาต่าง ๆ.....	37

สารบัญภาพ

ภาพที่	หน้า
2-1 การทำงานของขั้นตอนวิธีเชิงพันธุกรรม.....	5
2-2 การเข้ารหัสแบบไบนารี (Binary Encoding).....	6
2-3 การเข้ารหัสค่าต่าง ๆ (Value Encoding).....	6
2-4 ตัวอย่างการคำนวณค่าความเหมาะสม.....	7
2-5 ตัวอย่างการไขว้เปลี่ยน.....	8
2-6 การกลายพันธุ์.....	9
2-7 ประชากรต้นแบบ.....	13
2-8 ตัวอย่างปัญหาหออ้อยโรด.....	14
2-9 ตัวอย่างการคำนวณค่าความเหมาะสมของปัญหากับดัก.....	15
2-10 การแสดงการคำนวณค่าความเหมาะสมของปัญหากับดัก.....	16
3-1 ค่าความน่าจะเป็นที่เกิดขึ้นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ.....	21
3-2 การปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่.....	26
3-3 การหาค่าคำตอบของการทดลองกับพารามิเตอร์ $p_{size}/2$, $p_{size}/3$, $p_{size}/4$ และ $p_{size}/5$	34
4-1 ผลการทดลองจากปัญหาจำนวนบิตหนึ่งมากที่สุด.....	39
4-2 ผลการทดลองจากปัญหาเรนคอมเม็ก.....	40
4-3 ผลการทดลองจากปัญหาหออ้อยโรด.....	42
4-4 ผลการทดลองจากปัญหากับดัก.....	45
4-5 ค่าความน่าจะเป็น และค่าความเหมาะสม.....	48

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) ถูกนำเสนอโดย Holland (1973, 1975) และกลายมาเป็นที่รู้จักอย่างกว้างขวางจากหนังสือของ Goldberg (1989) เป็นขั้นตอนวิธีที่ใช้ในการค้นหาคำตอบ ด้วยวิธีการเลียนแบบการวิวัฒนาการของสิ่งมีชีวิตในธรรมชาติ โดยใช้ข้อมูลจากกลุ่มประชากรในการค้นหาคำตอบด้วยวิธีทางพันธุกรรม เช่น การไขว้เปลี่ยน และการกลายพันธุ์ เพื่อให้ได้ประชากรรุ่นใหม่ที่มีวิวัฒนาการที่ดีกว่าและมีความเหมาะสมมากยิ่งขึ้น

ขั้นตอนวิธีเชิงพันธุกรรมได้ถูกนำไปใช้ และมีการพัฒนาขั้นตอนวิธีการทำงานเรื่อยมา จนในปี ค.ศ. 1999 ขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับ (Compact Genetic Algorithm) ได้ถูกนำเสนอ โดย Harik et al. ซึ่งมีการทำงานที่แตกต่างจากขั้นตอนวิธีเชิงพันธุกรรมแบบเดิม โดยขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับมีแนวคิดในการใช้ตัวแบบความน่าจะเป็น (Probability Model) แทนการจัดเก็บกลุ่มประชากร และไม่จำเป็นต้องใช้การดำเนินการทางพันธุกรรม ทำให้ขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับมีการทำงานที่ใช้พื้นที่ในหน่วยความจำน้อย

การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับนั้น ใช้เวกเตอร์ความน่าจะเป็น (Probability Vector) ในการแทนคำตอบ ซึ่งขนาดของเวกเตอร์ความน่าจะเป็นมีขนาดเท่ากับความยาวของโครโมโซม และค่าความน่าจะเป็นแต่ละค่าในเวกเตอร์ความน่าจะเป็นคือค่าความน่าจะเป็นที่จะเกิดบิตที่มีค่า 1 หากค่าความน่าจะเป็นมีค่ามาก โอกาสที่จะเกิดบิต 1 ก็จะมีมาก แต่ถ้าหากค่าความน่าจะเป็นมีค่าน้อย โอกาสที่จะเกิดบิต 1 ก็จะน้อยลงไปด้วย

ถึงแม้ว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับจะมีการทำงานที่ใช้พื้นที่ในหน่วยความจำที่น้อย และไม่จำเป็นต้องอาศัยการดำเนินการทางพันธุกรรมแล้วก็ตาม แต่ในการทำงานกับปัญหาที่ยาก และซับซ้อนนั้น ขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับก็ยังไม่สามารถแก้ไขปัญหา และหาคำตอบที่ดีที่สุดได้

ดังนั้นในงานวิทยานิพนธ์นี้ ผู้วิจัยจึงได้ทำการศึกษาขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับ และปรับปรุงวิธีการปรับค่าเวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับให้มีประสิทธิภาพเพิ่มขึ้น

วัตถุประสงค์ของการวิจัย

เพื่อปรับปรุงวิธีการปรับค่าเวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบ กระชับให้มีประสิทธิภาพเพิ่มขึ้น

แนวทางในการศึกษา วิเคราะห์ และออกแบบอัลกอริทึม

1. ศึกษาขั้นตอนวิธีเชิงพันธุกรรม และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ
2. ศึกษาปัญหาต่าง ๆ เพื่อนำมาทดสอบกับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ได้แก่ ปัญหาจำนวนบิตหนึ่งมากที่สุด (One-Max Problem), ปัญหาแรนคอมแม็ก (Random Max Problem), ปัญหารอยัลโรด (Royal Road Problem) และปัญหาค้าง (Trap Problem)
3. ศึกษาขั้นตอนวิธีที่มีผู้อื่นนำเสนอมาก่อนหน้านี้ ได้แก่ ขั้นตอนวิธีเชิงพันธุกรรมแบบ กระชับด้วยค่าเฉลี่ย (mcGA), ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับโดยใช้ตัวเก่ง แบบถาวร (pe-cGA) และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับโดยใช้ตัวเก่งแบบชั่วคราว (ne-cGA)
4. ออกแบบวิธีการแก้ปัญหา เพื่อปรับปรุงวิธีการปรับค่าเวกเตอร์ความน่าจะเป็นของ ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มขึ้น
5. ทำการทดลองขั้นตอนวิธี และเปรียบเทียบผลการทดลองกับขั้นตอนวิธีที่มีผู้อื่น นำเสนอมาก่อนหน้านี้

ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

สามารถปรับปรุงขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มมากขึ้น และสามารถแก้ปัญหาที่ยาก เช่น ปัญหาค้างได้อย่างมีประสิทธิภาพเพิ่มมากขึ้น

ขอบเขตของการวิจัย

ในการทดลองจะทดสอบวิธีการที่นำเสนอกับปัญหาที่ถูกออกแบบมาเพื่อทดสอบ ประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรม ได้แก่ ปัญหาจำนวนบิตหนึ่งมากที่สุด ปัญหาแรนคอมแม็ก ปัญหารอยัลโรด และปัญหาค้าง

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

งานวิจัยนี้ผู้วิจัยได้ศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้องต่าง ๆ เพื่อนำมาใช้เป็นความรู้ในการพัฒนาขั้นตอนวิธีในการปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกะทัดรัด โดยผู้วิจัยได้ทำการพัฒนาโปรแกรมตามทฤษฎีที่ศึกษามา และทำการทดลองเพื่อศึกษาถึงปัญหาที่เกิดขึ้น ซึ่งทฤษฎี และงานวิจัยที่เกี่ยวข้องมีดังนี้

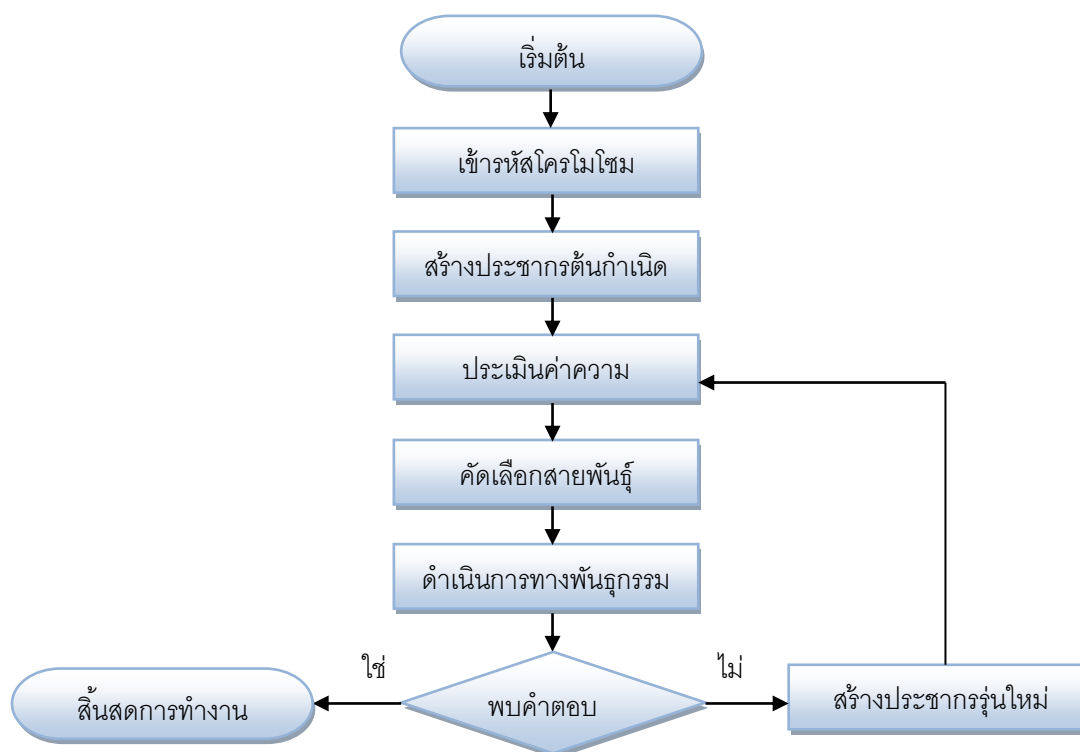
1. ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm)
 - 1.1. การทำงานของขั้นตอนวิธีเชิงพันธุกรรม
 - 1.1.1. การเข้ารหัสโครโมโซม
 - 1.1.2. สุ่มสร้างประชากรต้นกำเนิด
 - 1.1.3. คำนวณค่าความเหมาะสมของประชากรแต่ละตัว
 - 1.1.4. ดำเนินการค้นหาคำตอบด้วยวิธีทางพันธุกรรม
2. ขั้นตอนวิธีเชิงพันธุกรรมแบบกะทัดรัด (Compact Genetic Algorithm)
 - 2.1. การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกะทัดรัด
3. ปัญหาที่ใช้ทดสอบ
 - 3.1. ปัญหาจำนวนบิตหนึ่งมากที่สุด (One-Max Problem)
 - 3.2. ปัญหาแรนดอมแม็กซ์ (Random Max Problem)
 - 3.3. ปัญหารอยัลโรด (Royal Road Problem)
 - 3.4. ปัญหากับดัก (Trap Problem)
4. งานวิจัยที่เกี่ยวข้อง

ขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนวิธีเชิงพันธุกรรมเป็นขั้นตอนวิธีที่ใช้ในการค้นหาคำตอบด้วยวิธีการเลียนแบบการวิวัฒนาการของสิ่งมีชีวิตในธรรมชาติ โดยใช้ข้อมูลจากกลุ่มประชากรในการค้นหาคำตอบด้วยวิธีทางพันธุกรรม เช่น การไขว้เปลี่ยน และการกลายพันธุ์ เพื่อให้ได้ประชากรรุ่นใหม่ที่มีวิวัฒนาการที่ดีมีความเหมาะสมมากยิ่งขึ้น โดยขั้นตอนการค้นหาคำตอบเริ่มขึ้นจากการสุ่มสร้างประชากรขึ้นมาซึ่งประชากรแต่ละตัวจะอยู่ในรูปแบบของโครโมโซมที่ประกอบด้วยบิต 0 หรือ 1 โดยหนึ่งประชากรอาจมีมากกว่า 1 โครโมโซม และประชากรหลาย ๆ ประชากรรวมกันกลายเป็นกลุ่มประชากร เมื่อได้ประชากรมาแล้วจึงทำการคำนวณค่าความเหมาะสมของประชากรแต่ละตัว และทำการคัดเลือกประชากรที่มีค่าความเหมาะสมที่ดีมาสร้างประชากรรุ่นใหม่ที่มีค่าความเหมาะสมมากยิ่งขึ้นโดยใช้ตัวดำเนินการทางพันธุกรรม ซึ่งรายละเอียดของขั้นตอนวิธีจะกล่าวถึงดังต่อไปนี้

1. การทำงานของขั้นตอนวิธีเชิงพันธุกรรม

การทำงานของขั้นตอนวิธีเชิงพันธุกรรมนั้นจะเริ่มจากการเข้ารหัสโครโมโซม แล้วจึงเริ่มต้นการสุ่มสร้างประชากรต้นกำเนิด เมื่อได้ประชากรต้นกำเนิดแล้วจึงทำการประเมินค่าความเหมาะสมของประชากรแต่ละตัว ซึ่งประชากรที่ดีในกลุ่มที่ถูกประเมินค่าแล้วอาจจะถูกคัดเลือก และดำเนินการทางพันธุกรรม เช่น การไขว้เปลี่ยน การกลายพันธุ์ เพื่อสร้างประชากรรุ่นใหม่ และจะวนซ้ำอยู่อย่างนี้จนกว่าจะได้ประชากรที่ตรงตามความต้องการ ดังภาพที่ 2-1



ภาพที่ 2-1 การทำงานของขั้นตอนวิธีเชิงพันธุกรรม

1.1. การเข้ารหัสโครโมโซม (Chromosome Encoding)

การเข้ารหัสโครโมโซมมีอยู่หลายรูปแบบ เช่น

- 1) การเข้ารหัสแบบไบนารี (Binary Encoding) หรือการเข้ารหัสด้วยเลขฐานสองเป็นการเข้ารหัสโครโมโซมแบบปรกติที่ใช้กันทั่วไป โดยในแต่ละยีนของโครโมโซมจะประกอบไปด้วยเลขฐานสอง คือ 0 และ 1 ดังตัวอย่างในภาพที่ 2-2 โดยทั่วไปและในงานวิจัยนี้การเข้ารหัสโครโมโซมจะอยู่ในรูปแบบเลขฐาน 2 ซึ่งหลังจากการเข้ารหัสโครโมโซมที่ได้จะประกอบไปด้วยตัวเลข 0 กับ 1 ความยาวของโครโมโซมจะขึ้นอยู่กับลักษณะของแต่ละปัญหา

โครโมโซมที่ 1: 1 0 0 1 1 1 1 0 0 0 1 1

โครโมโซมที่ 2: 1 0 0 0 0 0 1 0 0 0 1 0

ภาพที่ 2-2 การเข้ารหัสแบบไบนารี (Binary Encoding)

- 2) การเข้ารหัสแบบค่าต่าง ๆ (Value Encoding) เป็นการเข้ารหัสด้วยค่าต่าง ๆ ตามความเหมาะสมกับแต่ละปัญหา โดยแต่ละตำแหน่งของยีนในโครโมโซมจะแทนค่าต่าง ๆ ด้วยรูปแบบ ตัวอักษร จำนวนจริง หรือคำสั่งต่าง ๆ ดังตัวอย่างในภาพที่ 2-3

การเข้ารหัสแบบที่ 1: A f g h o r k m c e j c

การเข้ารหัสแบบที่ 2: 1.8 1.5 2.0 5.1 2.5 5.9

การเข้ารหัสแบบที่ 3 : Back left right left left

ภาพที่ 2-3 การเข้ารหัสค่าต่าง ๆ (Value Encoding)

1.2. สุ่มสร้างประชากรต้นกำเนิด

เป็นการสร้างประชากรชุดแรก หรือประชากรเริ่มต้น โดยการสุ่มขึ้นมาจากคำตอบที่เป็นไปได้ โดยทำการสุ่มขึ้นมาตามจำนวนของประชากรที่กำหนดไว้ของขั้นตอนวิธี เพื่อนำมาใช้เป็นจุดเริ่มต้นของขั้นตอนวิธีเชิงพันธุกรรม

1.3. กำหนดค่าความเหมาะสมของประชากรแต่ละตัว

เป็นการประเมินค่าความเหมาะสม (Fitness Value) ของประชากรแต่ละตัวเพื่อให้คะแนนสำหรับประชากรแต่ละตัวที่สร้างขึ้นมา โดยประชากรแต่ละตัวจะมีค่าความเหมาะสมของตัวเอง เพื่อนำไปใช้ในการประเมินว่าประชากรตัวใดมีความเหมาะสมที่จะนำไปใช้ในการสืบทอดพันธุกรรม โดย

วิธีการประเมินค่าความเหมาะสมจะขึ้นอยู่กับลักษณะของแต่ละปัญหา เช่น สมมติว่าฟังก์ชันวัตถุประสงค์ (Objective Function) คือ $f(x) = x^2$

สมมติว่าโครโมโซมมีการเข้ารหัสที่เป็นคำตอบด้วยเลขฐาน 2 จำนวน 8 บิต การหาค่าความเหมาะสมจะคำนวณจากการถอดรหัสโครโมโซมที่เป็นเลขฐานสองให้เป็นเลขฐานสิบ แล้วแทนค่า x ลงในสมการดังภาพที่ 2-4

$$\text{Decode} \Rightarrow \text{โครโมโซม: } \underline{0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0} \Rightarrow \text{Fitness } f(x) = 22^2$$

ภาพที่ 2-4 ตัวอย่างการคำนวณค่าความเหมาะสม

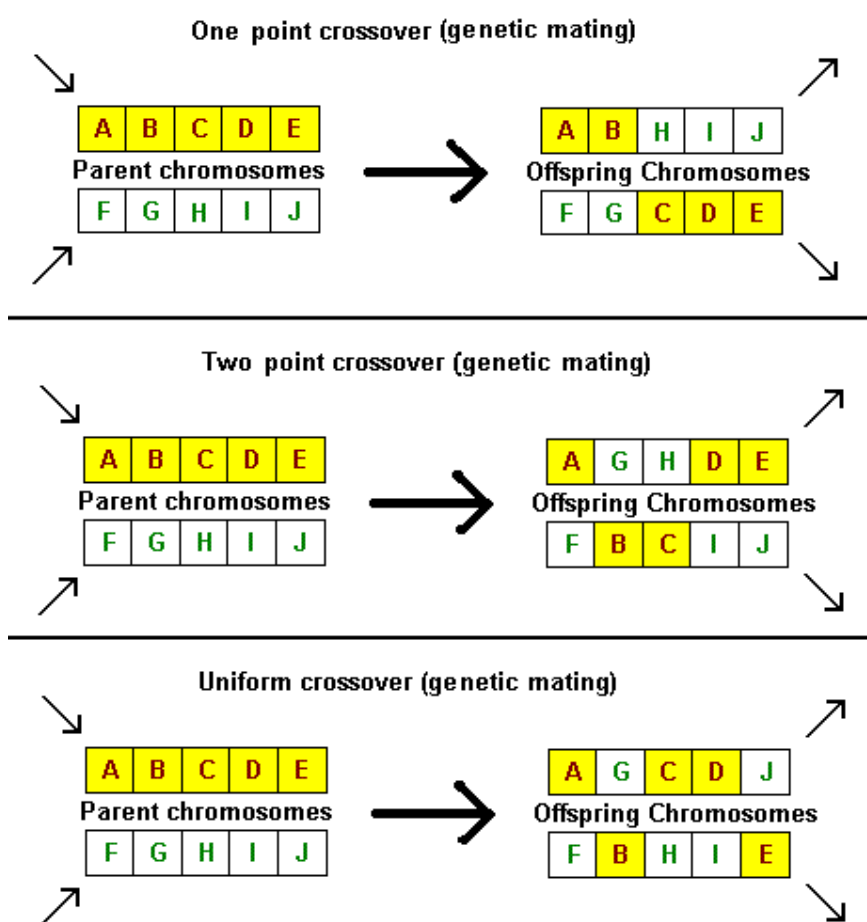
1.4. ดำเนินการค้นหาคำตอบด้วยวิธีทางพันธุกรรม

เป็นการดำเนินการเพื่อค้นหาคำตอบตามขั้นตอนวิธีเชิงพันธุกรรม เพื่อให้ได้ประชากรรุ่นใหม่ที่มีวิวัฒนาการของคำตอบที่ดีขึ้น โดยขั้นตอนวิธีทางพันธุกรรมประกอบไปด้วย การคัดเลือกสายพันธุ์ การไขว้เปลี่ยน และการกลายพันธุ์

การคัดเลือกสายพันธุ์ (Selection) คือการคัดเลือกประชากรที่มีค่าความเหมาะสมที่ดีกว่ามาใช้ในการสร้างประชากรรุ่นใหม่ เทคนิคที่นิยมใช้ในการคัดเลือก เช่น การคัดเลือกชนิดวงล้อรูเล็ต (Roulette Wheel Selection) และการคัดเลือกโดยการแข่งขัน (Tournament Selection) เป็นต้น

การไขว้เปลี่ยน (Crossover) คือการผสมโครโมโซมที่ผ่านกระบวนการคัดเลือกเพื่อให้ได้ประชากรในรุ่นต่อไป โดยทั่วไปจะทำการเลือกโครโมโซมตัวอย่างออกมา 1 คู่ จากนั้นทำการไขว้เปลี่ยนระหว่างโครโมโซมซึ่งจะได้โครโมโซมใหม่ออกมาสองโครโมโซมและทำซ้ำจนได้จำนวนโครโมโซมครบตามที่เรากำลังต้องการ (ครบตามจำนวนประชากรที่กำหนด) โดยทั่วไปการไขว้เปลี่ยนจะไม่ได้เกิดในทุกโครโมโซมแต่จะมีความน่าจะเป็นในการไขว้เปลี่ยนเพื่อจะบอกว่าโครโมโซมใดจะถูกไขว้เปลี่ยนกันหรือไม่และส่วนโครโมโซมที่ไม่ได้รับการไขว้เปลี่ยนโครโมโซมนั้นจะถูกคัดลอกไปเป็นประชากรรุ่นใหม่ที่มีเหมือนต้นแบบทุกประการ ตัวอย่างของการไขว้เปลี่ยนแสดงในภาพที่ 2-5 การ

เข้ารหัสโดยทั่วไปจะเป็นระบบเลขฐานสอง จึงจะขอยกตัวอย่างเป็นการเข้ารหัสของระบบเลขฐานสอง การแลกเปลี่ยนยีนจะเริ่มจากการกำหนดจำนวนจุดที่จะทำการไขว้เปลี่ยน โดยต้องมีจำนวนจุดมากกว่า หรือเท่ากับ 1 จุด กรณีที่จุดที่มีการไขว้เปลี่ยนมี 1 จุด จะทำการไขว้เปลี่ยนทั้งหมดหลังจุดที่กำหนดกรณีของการไขว้เปลี่ยนแบบ 2 จุด จะทำการไขว้เปลี่ยนระหว่างจุดทั้งสอง และการไขว้เปลี่ยนแบบ uniform จะแลกเปลี่ยนข้อมูลในแต่ละยีน



ภาพที่ 2-5 ตัวอย่างการไขว้เปลี่ยน (อ้างอิงจาก:www.softtechdesign.com/GA/EvolvingABetterSolution-GA.html)

การกลายพันธุ์ (Mutation) เป็นการสร้างโครโมโซมใหม่จากโครโมโซมเดิม ซึ่งจะส่งผลให้โครโมโซมที่ถูกสร้างขึ้นใหม่ด้วยตัวดำเนินการนี้จะถูกเปลี่ยนแปลงยีนบางตำแหน่งให้เป็นค่าใหม่ ดังภาพที่ 2-6

ยีนกลายพันธุ์

⇓

โครโมโซมต้นแบบ :	1	1	0	0	1	1	1	0	1	0
โครโมโซมกลายพันธุ์ :	1	1	1	0	1	1	1	0	1	0

ภาพที่ 2-6 การกลายพันธุ์

ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ เป็นขั้นตอนวิธีเชิงพันธุกรรมที่มีการทำงานที่รวดเร็ว และใช้พื้นที่หน่วยความจำน้อย เนื่องจากขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับเป็นขั้นตอนวิธีที่ใช้เวกเตอร์ความน่าจะเป็นแทนการจัดเก็บกลุ่มประชากรแบบขั้นตอนวิธีเชิงพันธุกรรมแบบเดิม จึงทำให้ใช้พื้นที่ในหน่วยความจำน้อย และมีการประมวลผลที่รวดเร็ว เนื่องจากไม่จำเป็นต้องใช้วิธีการทางพันธุกรรม เช่น การไขว้เปลี่ยน หรือ การกลายพันธุ์ โดยเวกเตอร์ความน่าจะเป็นจะประกอบไปด้วยมิติที่มีจำนวนเท่ากับความยาวของโครโมโซม และแต่ละมิติของเวกเตอร์ความน่าจะเป็นจะประกอบไปด้วยค่าความน่าจะเป็น 0.0 ถึง 1.0 ซึ่งค่าความน่าจะเป็นนี้คือค่าความน่าจะเป็นที่จะเกิดบิตที่เป็น 1 ในแต่ละโครโมโซมของประชากร เช่น ถ้าหากในมิติแรกของเวกเตอร์ความน่าจะเป็นมีค่าเป็น 0.5 นั่นก็แสดงว่าในบิตแรกของประชากรมีความน่าจะเป็นที่จะเป็น 1 หรือ 0 อย่างละครึ่งนั่นเอง

ในการค้นหาคำตอบที่มีความเหมาะสมมากที่สุด ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะทำการปรับปรุงค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นตามประชากรที่มีค่าความ

เหมาะสมที่สุดว่า โดยในการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะเริ่มจากการสุ่มสร้างประชากรขึ้นมาสองตัว โดยใช้ค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นในการสร้างประชากร ซึ่งแต่ละบิตของประชากรที่ได้จะมีค่าเป็น 0 หรือ 1 โดยขึ้นอยู่กับค่าความน่าจะเป็นของแต่ละมิติของเวกเตอร์ความน่าจะเป็นในมิตินั้น ๆ เช่น ถ้าหากค่าความน่าจะเป็นในมิตินั้น ๆ เป็น 0.5 ก็แสดงว่าในบิตนั้นของประชากรก็มีโอกาสที่จะเป็น 0 หรือ 1 เท่ากัน แต่ถ้าหากค่าความน่าจะเป็นในมิตินั้น ๆ เป็น 1.0 ก็แสดงว่าในบิตนั้นของประชากรก็มีโอกาสที่จะเป็น 1 อย่างแน่นอน

เมื่อทำการสุ่มสร้างประชากรเรียบร้อยแล้ว ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะทำการคำนวณค่าความเหมาะสมของประชากรแต่ละประชากรเพื่อหาประชากรที่มีค่าความเหมาะสมที่สุดว่า เมื่อได้ประชากรที่มีความเหมาะสมที่สุดแล้ว ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะทำการปรับปรุงค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นตามประชากรที่มีค่าความเหมาะสมที่สุดว่า โดยถ้าหากในแต่ละบิตของประชากรที่มีค่าความเหมาะสมดีว่ามีค่าเป็น 1 แต่ของประชากรที่มีค่าความเหมาะสมน้อยกว่าเป็น 0 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะทำการบวกค่าความน่าจะเป็นปรับเพิ่มเข้าไปในมิตินั้น แต่ถ้าหากในแต่ละบิตของประชากรที่มีค่าความเหมาะสมดีว่ามีค่าเป็น 0 แต่ของประชากรที่มีค่าความเหมาะสมน้อยกว่าเป็น 1 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะทำการปรับลดค่าความน่าจะเป็นในมิตินั้น

1. การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ l คือ ความยาวโครโมโซม p คือ เวกเตอร์ความน่าจะเป็น $p[i]$ คือ ค่าความน่าจะเป็นที่แต่ละบิตจะเป็น 1

1.1. กำหนดค่าความน่าจะเป็นเริ่มต้นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นเป็น 0.5

for $i := 1$ to l do

$p[i] := 0.5;$

1.2. สุ่มสร้างประชากรโดยใช้เวกเตอร์ความน่าจะเป็นขึ้นมา 2 ประชากร

$individual1 := generate(p);$

$individual2 := generate(p);$

1.3. คำนวณค่าความเหมาะสมของประชากรแต่ละตัว

1.4. เปรียบเทียบค่าความเหมาะสมของประชากรแต่ละตัวเพื่อหาผู้ชนะ

$winner, loser := compete(individual1, individual2);$

1.5. ปรับปรุงค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นตามประชากรผู้ชนะ โดยการบวกเพิ่ม หรือลดค่าความน่าจะเป็นในแต่ละมิติด้วยค่า $1/psize$ เมื่อ $psize$ คือขนาดของประชากรเมื่อเทียบกับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย

for $i := 1$ to l

begin

if $winner[i] \neq loser[i]$ then

if $winner[i] = 1$ then

$p[i] := p[i] + 1/psize;$

else

$p[i] := p[i] - 1/psize;$

end if

end if

end

1.6. ทำซ้ำตามขั้นตอนที่ 1.2 ถึง 1.5 จนกว่าค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นจะเป็น 0.0 หรือ 1.0 ทุกมิติ

ปัญหาที่ใช้ทดสอบ

1. ปัญหาจำนวนบิตหนึ่งมากที่สุด

ปัญหาจำนวนบิตหนึ่งมากที่สุดเป็นปัญหาง่ายที่ใช้ทดสอบขั้นตอนวิธีเชิงพันธุกรรม คำตอบที่ดีที่สุดของปัญหานี้คือทุกบิตมีค่าเป็น 1 ทั้งหมด ค่าความเหมาะสมของปัญหานี้คำนวณได้จากการนับจำนวนบิตที่เป็น 1 ของประชากรแต่ละตัว โดยยังมีจำนวนบิตที่เป็น 1 มากเท่าใด ก็ยังมีค่าความเหมาะสมดีมากขึ้น

ปัญหาจำนวนบิตหนึ่งมากที่สุดเป็นปัญหาที่ได้รับความนิยมไปใช้ในงานวิจัยด้านขั้นตอนวิธีเชิงพันธุกรรมหลาย ๆ งาน เนื่องจากปัญหาจำนวนบิตหนึ่งมากที่สุดเป็นปัญหาที่สามารถเข้าใจได้ง่าย วิเคราะห์ได้ง่ายซึ่งตัวอย่างการคำนวณค่าความเหมาะสมของปัญหาจำนวนบิตหนึ่งมากที่สุด แสดงในตารางที่ 2-1

ตารางที่ 2-1 ตัวอย่างปัญหาจำนวนบิตหนึ่งมากที่สุด

ประชากร										ค่าความเหมาะสม
1	1	0	0	1	0	0	0	0	1	4
1	0	1	1	0	1	0	1	0	0	5
1	0	1	1	0	1	0	1	0	1	6
1	1	0	1	1	1	0	1	1	0	7
1	1	1	1	1	0	1	1	1	0	8
1	1	1	1	1	1	1	1	1	0	9
1	1	1	1	1	1	1	1	1	1	10

2. ปัญหาแรนคอมแม็ก

ปัญหาแรนคอมแม็กเป็นปัญหาที่มีการสุ่มสร้างประชากรต้นแบบที่มีคำตอบที่ดีที่สุดขึ้นมา ก่อนแล้วจึงสุ่มสร้างประชากรในแต่ละรุ่นขึ้นมา โดยคำตอบที่ดีที่สุดของปัญหานี้คือทุก ๆ บิตของประชากรจะต้องมีค่าที่ตรงกับแต่ละบิตของประชากรต้นแบบ หากในบิตใดมีค่าไม่ตรงกับบิตนั้นในประชากรต้นแบบ ค่าความเหมาะสมในบิตนั้นจะมีค่าเป็น 0 กล่าวคือยังมีจำนวนบิตที่มีค่าตรงกับประชากรต้นแบบมากเท่าไร ค่าความเหมาะสมก็ยิ่งมากขึ้นเท่านั้น

จากที่กล่าวมาสมมติว่าเราได้ทำการสุ่มสร้างประชากรต้นแบบที่มีคำตอบที่ดีที่สุดขึ้นมาได้แล้ว

ประชากรต้นแบบ : 1 0 1 0 1 0 1 1 1 0

ภาพที่ 2-7 ประชากรต้นแบบ

การคำนวณค่าความเหมาะสมตัวอย่างดังตารางที่ 2-2

ตารางที่ 2-2 ตัวอย่างปัญหาแรนคอมเม็กซ์

ประชากร	ค่าความเหมาะสม
1 1 1 1 1 1 1 0 0 1	4
1 1 1 1 1 1 1 1 0 1	5
1 1 1 1 1 1 1 1 1 1	6
1 1 1 1 1 1 1 1 1 0	7
1 1 1 1 1 0 1 1 1 0	8
1 0 1 1 1 0 1 1 1 0	9
1 0 1 0 1 0 1 1 1 0	10

3. ปัญหาแรนคอมเม็กซ์

ปัญหาแรนคอมเม็กซ์เป็นปัญหาที่แต่ละบิตในโครโมโซมมีความสัมพันธ์กัน โดยที่โครโมโซมจะถูกแบ่งออกเป็นชุด และแต่ละชุดจะมีจำนวนของบิตที่เท่ากัน ซึ่งในที่นี้เรียกว่าบล็อก ค่าความเหมาะสมของโครโมโซมคือ ผลรวมของค่าความเหมาะสมในแต่ละบล็อกโดยค่าความเหมาะสมที่ได้มีค่าเท่ากับจำนวนของบิตในบล็อก ซึ่งจะต้องมีค่าเป็น 1 ทั้งหมด แต่ถ้าหากมีบิตใดบิตหนึ่งในบล็อกมีค่าเป็น 0 ค่าความเหมาะสมในบล็อกนั้นก็จะเป็น 0 ทันที ด้วยเหตุนี้ปัญหาแรนคอมเม็กซ์จึงเป็นปัญหาที่ยากสำหรับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

ภาพที่ 2-8 คือตัวอย่างปัญหาการยัลโรคขนาด 32 บิต ในขั้นตอนแรกเราทำการแบ่งโครโมโซมออกเป็นชุด ชุดละ 8 บิต ดังรูปแบบที่ 1 ถึง 4 และทำการคำนวณค่าความเหมาะสมของแต่ละชุด

หลังจากนั้นเราทำการแบ่งโครโมโซมออกเป็นชุด ชุดละ 16 บิต ซึ่งสามารถแบ่งได้ทั้งหมด 2 ชุด ดังรูปแบบที่ 5 ถึง 6 และทำการคำนวณค่าความเหมาะสมของแต่ละชุดออกมา

ในขั้นตอนสุดท้ายเราทำการแบ่งโครโมโซมได้เพียงแค่ชุดเดียว คือ 32 บิต ดังรูปแบบที่ 7 และทำการคำนวณค่าความเหมาะสม

เมื่อได้ค่าความเหมาะสมของแต่ละชุดตามขั้นตอนที่กล่าวมาแล้ว จึงนำค่าความเหมาะสมที่ได้ในแต่ละชุดมารวมกัน ซึ่งค่าความเหมาะสมที่ได้คือ ค่าความเหมาะสมเท่ากับ 8 ทั้งหมด 4 บล็อก ค่าความเหมาะสมเท่ากับ 16 ทั้งหมด 2 บล็อก และค่าความเหมาะสมเท่ากับ 32 ทั้งหมด 1 บล็อก หรือ $(8 \times 4) + (16 \times 2) + (32 \times 1) = 96$ ซึ่งเป็นคำตอบที่ดีที่สุด

รูปแบบที่ 1 = 1111111*****; ค่าความเหมาะสมเท่ากับ 8

รูปแบบที่ 2 = *****1111111*****; ค่าความเหมาะสมเท่ากับ 8

รูปแบบที่ 3 = *****1111111*****; ค่าความเหมาะสมเท่ากับ 8

รูปแบบที่ 4 = *****1111111; ค่าความเหมาะสมเท่ากับ 8

รูปแบบที่ 5 = 11111111111111*****; ค่าความเหมาะสมเท่ากับ 16

รูปแบบที่ 6 = *****11111111111111; ค่าความเหมาะสมเท่ากับ 16

รูปแบบที่ 7 = 111111111111111111111111111111; ค่าความเหมาะสมเท่ากับ 32

ภาพที่ 2-8 ตัวอย่างปัญหาการยัลโรค

4. ปัญหาทับคัท

ปัญหาทับคัทเป็นปัญหาที่ขั้นตอนวิธีเชิงพันธุกรรมหาคำตอบที่เหมาะสมที่สุดได้ยาก โดยปัญหาทับคัทจะประกอบไปด้วยชุดของจำนวนของบิตที่ต่อเข้าด้วยกันจนกลายเป็นทับคัทที่ยาวขึ้น

การหาค่าความเหมาะสมในปัญหาทับคัทนั้น จะทำการหาค่าความเหมาะสมในแต่ละชุดก่อน แล้วนำค่าความเหมาะสมที่ได้ทั้งหมดในแต่ละชุดมารวมกันจึงจะได้ค่าความเหมาะสมของประชากรตัวนั้นดังสมการที่ 2.1 และที่ 2.2

$$F_k(b_0 \dots b_{k-1}) = \begin{cases} f_{\text{high}} & ; \text{ if } u = k \\ f_{\text{low}} - u \frac{f_{\text{low}}}{k-1} & ; \text{ otherwise} \end{cases} \quad (2.1)$$

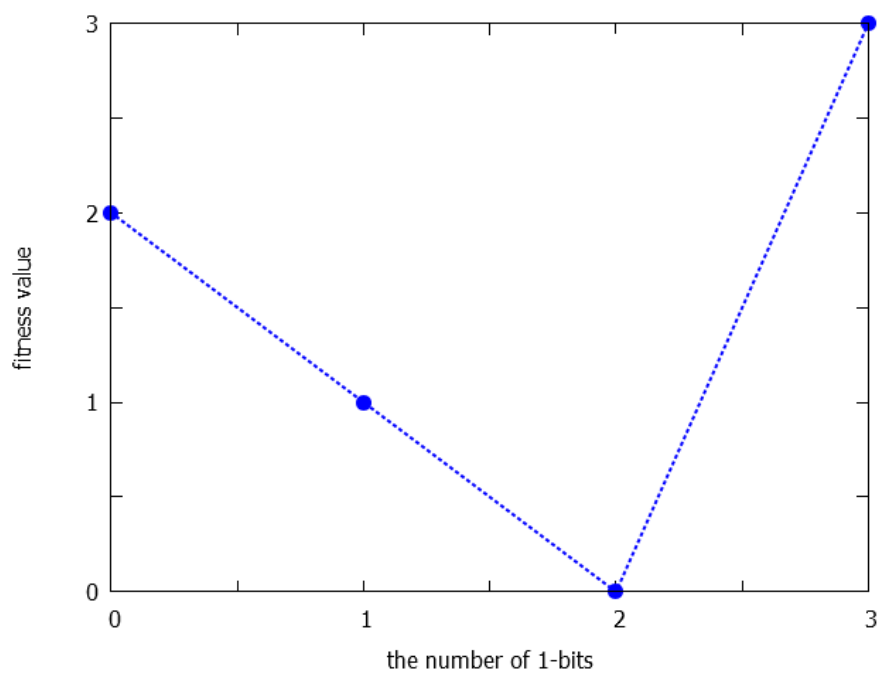
$$F_{k \times m}(B_0 \dots B_{m-1}) = \sum_{i=0}^{m-1} F_k(B_i), B_i \in \{0,1\}^k \quad (2.2)$$

เมื่อ $b_i \in \{0, 1\}$, $u = \sum_{i=0}^{k-1} b_i$ และ f_{high} คือ ค่าความเหมาะสม เมื่อทุกบิตในชุดเป็น 1 ทั้งหมด และ f_{low} คือ ค่าความเหมาะสมเมื่อทั้งชุดเป็น 0 ทั้งหมด โดย $f_{\text{high}} > f_{\text{low}}$ และ k คือความยาวของกลุ่มบิต ซึ่งมีความยาวเป็น k บิต จากสมการ f_{high} จะมีค่าเท่ากับ k ก็ต่อเมื่อ $u = k$ และ f_{low} มีค่าเท่ากับ $k-1$

จากสมการข้างต้น การนำค่าความเหมาะสมของแต่ละกลุ่มบิตขนาด k บิต มาต่อกันเป็น m หน่วย แสดงดังตัวอย่างในภาพที่ 2-9, 2-10 และตารางที่ 2-3 ต่อไปนี้

โครโมโซม	110	010	001	111	000						
ค่าความเหมาะสม	0	+	1	+	1	+	3	+	2	=	7

ภาพที่ 2-9 ตัวอย่างการคำนวณค่าความเหมาะสมของปัญหาทับคัท ขนาด 3



ภาพที่ 2-10 ค่าความเหมาะสมของปัญหากับดัก ขนาด 3

ตารางที่ 2-3 ตัวอย่างปัญหากับดัก

	ประชากร				ค่าความเหมาะสม
110	010	001	111	011	5
100	010	001	111	011	6
110	010	001	111	000	7
100	010	001	111	000	8
100	010	001	111	000	9
100	010	111	111	000	10
111	111	111	111	111	15

งานวิจัยที่เกี่ยวข้อง

Harik et al. (1999) นำเสนอขั้นตอนวิธีใหม่ในการทำงานของขั้นตอนวิธีเชิงพันธุกรรม เรียกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับ (Compact Genetic Algorithm) โดยนำเสนอการใช้ตัวแบบความน่าจะเป็น (Probability Vector) แทนกลุ่มประชากรในการหาคำตอบ ซึ่งทำให้สามารถลดการใช้หน่วยความจำจากการจัดเก็บกลุ่มประชากรได้ เนื่องจากวิธีนี้จะทำการจัดเก็บเพียงตัวแบบความน่าจะเป็นเท่านั้น และวิธีนี้ยังทำให้การประมวลผลเป็นไปได้อย่างรวดเร็ว เนื่องจากไม่จำเป็นต้องอาศัยการดำเนินการเชิงพันธุกรรม เช่น การไขว้เปลี่ยน และการกลายพันธุ์ โดยที่ยังคงความสามารถในการหาคำตอบได้เทียบเท่ากับขั้นตอนวิธีแบบเดิม

Sunisa Rimcharoen et al. (2006) นำเสนอขั้นตอนวิธีใหม่ในการปรับปรุงการปรับค่าของเวกเตอร์ความน่าจะเป็นในขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับ โดยการใช้ค่าเฉลี่ยของค่าความน่าจะเป็นในการปรับปรุงค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็น (Moving Average Compact Genetic Algorithm: mcGA) โดยในการทดลองได้ทำการทดลองกับ window size ขนาด 2 3 4 5 10 และ 15 ซึ่งผลการทดลองที่แสดงเป็นผลการทดลองที่ window size 10 จากผลการทดลองแสดงให้เห็นว่าวิธีการนี้สามารถเพิ่มประสิทธิภาพการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับในส่วนของคุณภาพคำตอบจำนวนครั้งในการประเมินค่าความเหมาะสมได้ เนื่องจากวิธีนี้ช่วยให้ชะลอการอัปเดตขึ้นลง ช่วยให้การปรับค่าความน่าจะเป็นที่ถูกต้องลดการตัดสินใจที่ผิดพลาด จนนำไปสู่การตัดสินใจที่ดีที่สุดในการปรับปรุงค่าความน่าจะเป็นไปในทิศทางที่มุ่งเข้าสู่คำตอบที่ดีที่สุด

Ahn and Ramakrishna (2003) นำเสนอขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับด้วยประชากรที่ดีที่สุดอย่างถาวร (Persistent Elitist Compact Genetic Algorithm: pe-cGA) และขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับด้วยประชากรที่ดีที่สุดไม่ถาวร (Nonpersistent Elitist Compact Genetic Algorithm: ne-cGA) โดยที่ pe-cGA เป็นขั้นตอนวิธีที่ทำการจัดเก็บประชากรตัวที่ดีที่สุดในปัจจุบันไว้ใช้จนกว่าจะได้คำตอบที่ดีที่สุด แต่ ne-cGA เป็นขั้นตอนวิธีที่ทำการจัดเก็บประชากรตัวที่ดีที่สุดไว้เพียงช่วงเวลาหนึ่งเท่านั้น จากผลการทดลองที่ได้ทั้งสองวิธีแสดงให้เห็นว่าสามารถเพิ่มประสิทธิภาพการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกะชับได้อย่างดีทั้งในส่วนของคุณภาพคำตอบจำนวนครั้งในการประเมินค่าความเหมาะสม

อรุวรรณ วัชฎาพร (2549) นำเสนอขั้นตอนวิธีเชิงพันธุกรรมคอมแพคท์แบบบีบอัด (Compressed Compact Genetic Algorithm – c²GA) เพื่อลดปริมาณการค้นหาคำตอบ ซึ่งเป็นการนำวิธีการเข้ารหัสแบบ CompressedGA มาใช้กับขั้นตอนวิธีเชิงพันธุกรรมคอมแพคท์ ทำให้สามารถค้นหาคำตอบได้เร็วกว่าขั้นตอนวิธีเชิงพันธุกรรมคอมแพคท์แบบดั้งเดิม โดยผู้ทดลองได้ทำการทดลองเปรียบเทียบระหว่างขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และขั้นตอนวิธีเชิงพันธุกรรมคอมแพคท์แบบบีบอัดกับปัญหาต่าง ๆ ได้แก่ ปัญหา OneMax, ปัญหา Royal Road, ปัญหา De Jong และปัญหาการโปรแกรมแขนหุ่นยนต์ ตามลำดับ ซึ่งผลการทดลองพบว่าขั้นตอนวิธีเชิงพันธุกรรมคอมแพคท์แบบบีบอัดสามารถแก้ปัญหาได้อย่างมีประสิทธิภาพมากกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

พินิจ อ่อนน้อย และสุพจน์ เสงพระพรหม (2557) นำเสนอการเพิ่มประสิทธิภาพขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยการปรับค่าเวกเตอร์ความน่าจะเป็น โดยค่านี้นักตามความเหมาะสม โดยผู้วิจัยได้ทำการกำหนดค่าที่ใช้ในการปรับเวกเตอร์ความน่าจะเป็นที่เหมาะสม โดยการใช้ค่าความแตกต่างของค่าความเหมาะสม (Fitness Value) ของประชากรทั้งสองตัวที่ถูกสุ่มสร้างขึ้นจากเวกเตอร์ความน่าจะเป็น ซึ่งในการทดลองผู้วิจัยได้ทำการทดลองกับปัญหาหนึ่งมากที่สุด (One-Max Problem) ซึ่งเป็นปัญหาพื้นฐานในการวัดประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรม จากการทดลองพบว่าขั้นตอนวิธีที่นำเสนอสามารถแก้ปัญหาได้เป็นอย่างดี โดยไม่จำเป็นต้องปรับค่าขนาดประชากรเพื่อให้เหมาะสมกับปัญหา

บทที่ 3

วิธีดำเนินงาน

การศึกษาขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

ในการดำเนินงาน ผู้วิจัย ได้ศึกษาการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยได้ทำการจัดเก็บข้อมูลค่าความน่าจะเป็นที่เกิดขึ้นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นตั้งแต่การทำงานรอบแรกจนถึงรอบสุดท้าย และนำผลการทดลองที่ได้มาศึกษา ซึ่งจากการศึกษาพบว่าขั้นตอนวิธีมีการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นที่แตกต่างกันออกไป บางบิตการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 หรือ 1.0 ได้อย่างต่อเนื่องและชัดเจน แต่ในหลาย ๆ บิตมีความไม่แน่นอนในการปรับปรุงค่าความน่าจะเป็นเกิดขึ้น ความไม่แน่นอนที่ว่านี้หมายถึงขั้นตอนวิธีมีการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 หรือ 0.1 สลับไปมา ซึ่งบางบิตก็สามารถที่จะหาทิศทางที่ชัดเจนได้ในตอนท้าย แต่บางบิตก็ไม่มีทิศทางที่ชัดเจนจนกระทั่งการทำงานรอบสุดท้าย และจากการศึกษาพบว่าขั้นตอนวิธีจะเริ่มมีทิศทางในการปรับปรุงค่าความน่าจะเป็นที่ชัดเจนก็ต่อเมื่อจำนวนของวิวัฒนาการได้เกิดขึ้นไปแล้วระยะหนึ่ง

ภาพที่ 3-1 เป็นผลการทดลองของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ที่ขนาดของ tournament เท่ากับ 2 และจำนวนประชากรมีขนาดเท่ากับ 20 ในภาพแสดงถึงค่าความน่าจะเป็นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นตั้งแต่การทำงานรอบแรกจนถึงรอบสุดท้าย โดยแนวตั้งคือรุ่นของวิวัฒนาการที่เกิดขึ้น แถวล่างสุดคือการทำงานรอบแรก และแถบบนสุดคือการทำงานรอบสุดท้าย แนวอนคือแต่ละบิตของเวกเตอร์ความน่าจะเป็น โดยนับบิตเริ่มต้นที่ 0 และโทนสีที่เกิดขึ้นนั้น โทนสีดำแสดงถึงค่าความน่าจะเป็น 0 และโทนสีขาวแสดงถึงค่าความน่าจะเป็น 1

จากภาพที่ 3-1 แสดงให้เห็นว่าขั้นตอนวิธีมีการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นที่แตกต่างกันออกไป โดยสังเกตได้จากโทนสีที่เกิดขึ้นในแต่ละบิตตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้าย ซึ่งในบางบิตมีการเปลี่ยนไปเปลี่ยนมาระหว่างโทนสีขาวและดำอย่างเห็นได้ชัด และมีรูปแบบที่แตกต่างกันไปในแต่ละบิต ดังแสดงให้เห็นในลูกศรที่ 1 ถึง 5 รายละเอียดดังนี้

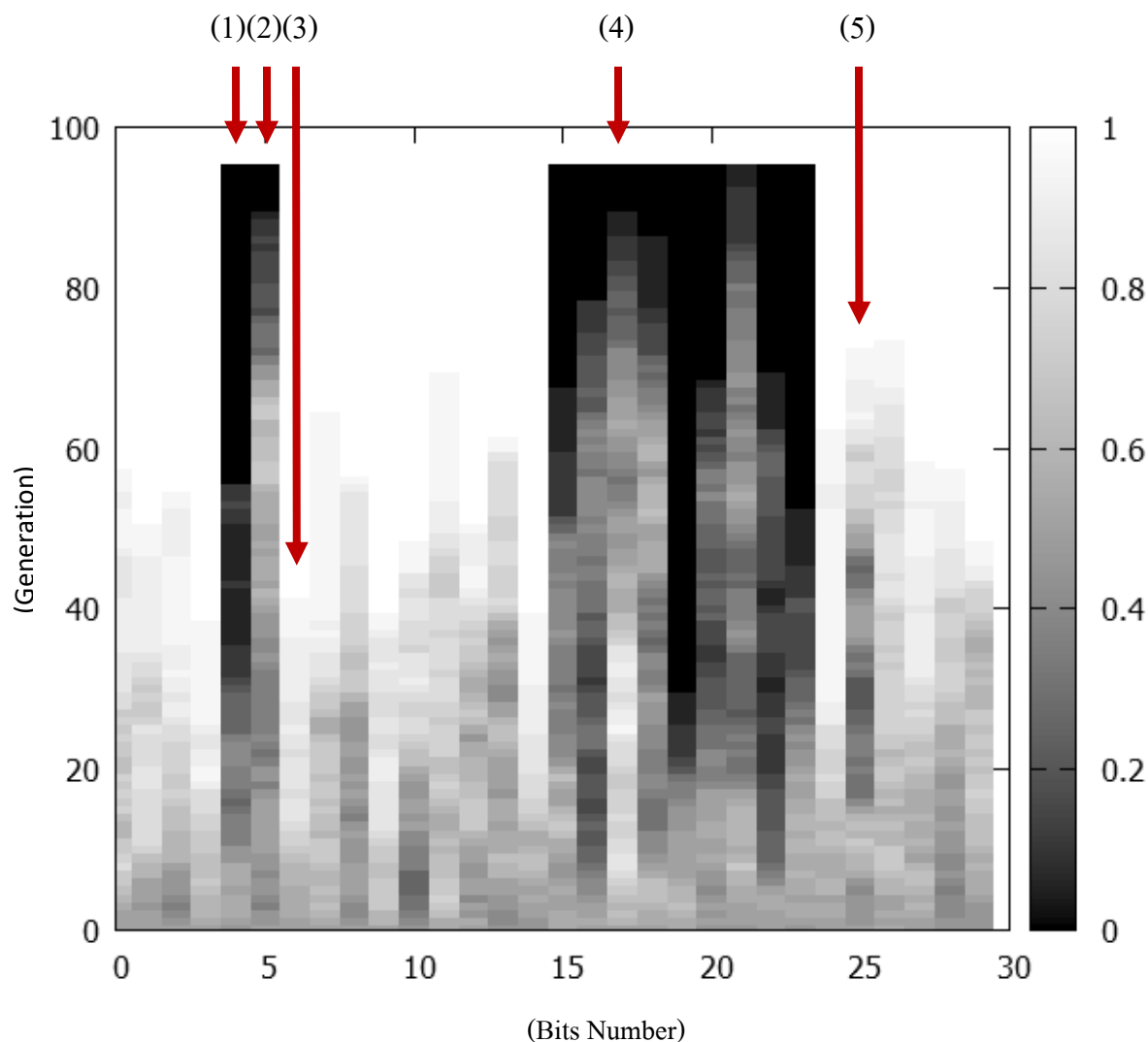
กฎศรที่ (1) แสดงถึงค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้ายของบิตที่ 4 ในเวกเตอร์ความน่าจะเป็น ซึ่งจะสังเกตได้ว่าในบิตนี้มีการปรับปรุงค่าความน่าจะเป็นไปในทิศทางที่แน่นอนและชัดเจนมาก โดยมีการเปลี่ยนแปลงของโทนสีจากโทนสีเทาไปหาโทนสีดำอย่างต่อเนื่อง และกลายเป็นสีดำในที่สุด ซึ่งสามารถที่จะบอกได้ว่าการปรับปรุงค่าความน่าจะเป็นในบิตนี้นั้นมีการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 อย่างต่อเนื่อง

กฎศรที่ (2) แสดงถึงค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้ายของบิตที่ 5 ในเวกเตอร์ความน่าจะเป็น ซึ่งจะสังเกตได้ว่าในบิตนี้มีความไม่แน่นอนเกิดขึ้นอยู่มาก โดยมีการเปลี่ยนไปเปลี่ยนมาระหว่างโทนสีขาวและดำอย่างเห็นได้ชัด และกว่าที่จะมีทิศทางที่แน่นอนก็เกือบจะถึงการทำงานรอบสุดท้ายแล้ว

กฎศรที่ (3) แสดงถึงค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้ายของบิตที่ 6 ในเวกเตอร์ความน่าจะเป็น ซึ่งจะสังเกตได้ว่าในบิตนี้มีการปรับปรุงค่าความน่าจะเป็นไปในทิศทางที่แน่นอนและชัดเจนมาก โดยมีการเปลี่ยนแปลงของโทนสีจากโทนสีเทาไปหาโทนสีขาวอย่างต่อเนื่อง และกลายเป็นสีขาวในที่สุด ซึ่งสามารถที่จะบอกได้ว่าการปรับปรุงค่าความน่าจะเป็นในบิตนี้นั้นมีการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 อย่างต่อเนื่อง

กฎศรที่ (4) แสดงถึงค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้ายของบิตที่ 17 ในเวกเตอร์ความน่าจะเป็น ซึ่งจะสังเกตได้ว่าในบิตนี้มีความไม่แน่นอนเกิดขึ้นมากเช่นกัน โดยในการทำงานช่วงแรกมีการเปลี่ยนแปลงของโทนสีจากโทนสีเทาไปหาโทนสีขาว แล้วเปลี่ยนไปหาโทนสีเทา และเปลี่ยนแปลงไปหาโทนสีดำในการทำงานรอบท้าย ๆ

กฎศรที่ (5) แสดงถึงค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้ายของบิตที่ 25 ในเวกเตอร์ความน่าจะเป็น ซึ่งจะสังเกตได้ว่าในบิตนี้มีความไม่แน่นอนเกิดขึ้นมากเช่นกัน โดยในการทำงานช่วงแรกมีโทนสีเทา แล้วเปลี่ยนไปหาโทนสีเทาเข้ม และเปลี่ยนกลับไปหาโทนสีขาวในการทำงานรอบท้าย ๆ



ภาพที่ 3-1 ค่าความน่าจะเป็นที่เกิดขึ้นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

จากการศึกษาการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของขั้นตอนวิธีนั้น พบว่ามีความถี่และความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 และ 0.0 เกิดขึ้น ซึ่งอาจจะเป็นปัจจัยที่ทำให้เกิดทิศทางที่แน่นอนในการปรับปรุงค่าความน่าจะเป็น ดังนั้นผู้วิจัยจึงได้ศึกษาค่าความถี่ และค่าความต่อเนื่องนี้ เพื่อนำมาใช้จัดการกับปัญหาความไม่แน่นอนในการปรับปรุงค่าความน่าจะเป็นให้สามารถตัดสินใจไปในทิศทางที่ถูกต้องได้ ซึ่งความหมายของค่าความถี่ และค่าความต่อเนื่องอธิบายดังต่อไปนี้

การศึกษาค่าความถี่ และค่าความต่อเนื่องในการปรับค่าของเวกเตอร์ความน่าจะเป็น

ค่าความถี่ คือ จำนวนครั้งในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 และ 0.0 ที่เกิดขึ้นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้าย ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับมีการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นในรอบการทำงาน ซึ่งแต่ละรอบการทำงานการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตจะมีความแตกต่างกัน บางรอบก็ปรับปรุงให้เข้าใกล้ค่า 1.0 บางรอบก็ปรับปรุงให้เข้าใกล้ค่า 0.0 และจากการศึกษาการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของขั้นตอนวิธีนั้น พบว่าในบิตที่มีการปรับปรุงเข้าหาค่า 1.0 อย่างต่อเนื่องจะมีจำนวนครั้งในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 มากกว่าค่า 0.0 และในบิตที่มีการปรับปรุงเข้าหาค่า 0.0 อย่างต่อเนื่องจะมีจำนวนครั้งในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 มากกว่าค่า 1.0 เช่นเดียวกัน ดังนั้นอาจนิยามได้ว่าถ้าหากบิตใดมีจำนวนครั้งในการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 มากกว่า 0.0 ก็อาจจะเป็นไปได้ว่าบิตนั้นมีแนวโน้มที่คำตอบจะเป็นบิต 1 แต่ถ้าหากมีจำนวนครั้งในการปรับปรุงค่าความน่าจะเป็นเข้าหา 0.0 มากกว่า 1.0 ก็อาจจะเป็นไปได้ว่าบิตนั้นมีแนวโน้มว่าคำตอบจะเป็นบิต 0

ค่าความต่อเนื่อง คือ จำนวนครั้งของการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 และ 0.0 ที่เป็นไปได้ในทิศทางเดียวกันอย่างต่อเนื่อง ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับมีการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นในรอบการทำงาน ซึ่งในแต่ละรอบการทำงานบางบิตอาจจะปรับเข้าหาค่า 1.0 บางบิตอาจจะปรับเข้าหาค่า 0.0 เมื่อการทำงานผ่านไปหลาย ๆ รอบการทำงานบางบิตมีการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 หรือ 0.0 ติดต่อกันหลายรอบการทำงาน ซึ่งแสดงถึงความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 หรือ 0.0 ในบิตนั้นนั่นเอง จากการศึกษาการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของขั้นตอนวิธีนั้น พบว่าในบิตที่มีทิศทางในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 หรือ 0.0 อย่างชัดเจนจะมีค่าความต่อเนื่องสูง แต่ในบิตที่มีความไม่แน่นอนเกิดขึ้นมากนั้นค่าความต่อเนื่องจะน้อยมาก

จากการศึกษาค่าความถี่ และค่าความต่อเนื่องที่เกิดขึ้นในแต่ละบิตของเวกเตอร์ความน่าจะเป็น โดยการนับจำนวนครั้งของความถี่ และความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าสู่ 1.0 (ปรับขึ้น) และทิศทางเข้าสู่ 0.0 (ปรับลด) พบว่าในแต่ละบิตนั้นมีค่าความถี่ และค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นที่แตกต่างกันออกไป ดังเช่นในบิตที่มีการปรับปรุงค่าความน่าจะเป็นไปในทิศทางเดียวกันอย่างต่อเนื่องนั้นจะมีค่าความถี่ และค่าความต่อเนื่องของการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 และ 0.0 ที่แตกต่างกันอย่างชัดเจน แต่ในบิตที่ยังมีความไม่แน่นอนของการปรับปรุงค่าความน่าจะเป็นอยู่มากนั้น ค่าความถี่ และค่าความต่อเนื่อง

ของการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 และ 0.0 จะมีความใกล้เคียงกัน แต่ก็ยังคงมีทิศทางไปในทิศทางเดียวกันกับผลลัพธ์ที่ได้ในบิตนั้น ๆ ซึ่งแสดงให้เห็นว่าความถี่ และความต่อเนื่องที่เกิดขึ้นในการปรับปรุงค่าความน่าจะเป็นสามารถที่จะบอกได้ถึงทิศทางในการปรับปรุงค่าความน่าจะเป็น ดังแสดงในตารางที่ 3-1

ตารางที่ 3-1 แสดงถึงค่าความถี่รวม และค่าความต่อเนื่องที่สูงที่สุดในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 และ 0.0 ตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้ายในบิตที่ 4, 5, 6, 17 และ 25 ของผลการทดลองจากภาพที่ 3-1 โดยแสดงให้เห็นว่าค่าความน่าจะเป็นในแต่ละบิตจากภาพที่ 3-1 นั้นมีทิศทางไปในทิศทางเดียวกันกับค่าความถี่ และค่าความต่อเนื่องที่เกิดขึ้นของบิตนั้น ๆ และแสดงให้เห็นว่าในบิตที่มีการปรับปรุงค่าความน่าจะเป็นไปในทิศทางเดียวกันอย่างต่อเนื่องนั้นจะมีค่าความถี่ และค่าความต่อเนื่องของการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 และ 0.0 ที่แตกต่างกันอย่างชัดเจน ดังเช่น ในบิตที่ 6 และในบิตที่ยังมีความไม่แน่นอนของการปรับปรุงค่าความน่าจะเป็นอยู่มากนั้น ค่าความถี่ และค่าความต่อเนื่องของการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 และ 0.0 ก็จะมีค่าใกล้เคียงกัน ดังเช่นในบิตที่ 17

ตารางที่ 3-1 ค่าความถี่ และค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 และ 0.0 ในบิตที่ 4, 5, 6, 17 และ 25 ของผลการทดลองจากภาพที่ 3-1

บิต	ค่าความถี่		ค่าความต่อเนื่องที่สูงที่สุด	
	เข้าหา 1.0	เข้าหา 0.0	เข้าหา 1.0	เข้าหา 0.0
บิตที่ 4	9	14	3	6
บิตที่ 5	22	26	3	5
บิตที่ 6	13	3	7	1
บิตที่ 17	22	29	5	6
บิตที่ 25	23	18	7	4

ดังนั้นผู้วิจัยจึงได้ทำการศึกษาถึงค่าความถี่ และค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นนี้ ซึ่งอาจจะสามารถนำมาใช้ในการปรับปรุงขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มมากขึ้น

และสามารถจัดการกับปัญหาความไม่แน่นอนในการปรับปรุงค่าความน่าจะเป็นให้สามารถตัดสินใจไปในทิศทางที่ถูกต้องได้

อัลกอริทึมที่นำเสนอ

จากการศึกษาค่าความถี่ และค่าความต่อเนื่องที่เกิดขึ้นในการปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ผู้วิจัยได้นำค่าความถี่ และค่าความต่อเนื่องมาใช้ในการปรับปรุงค่าความน่าจะเป็นในการทำงานแต่ละรอบของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยได้ทำการจัดเก็บข้อมูลค่าความถี่ และค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นที่เกิดขึ้นในแต่ละบิตของเวกเตอร์ความน่าจะเป็นในทุกรอบการทำงาน แล้วนำค่าความถี่มาใช้ในการตัดสินใจ และนำค่าความต่อเนื่องมาใช้ในการปรับปรุงค่าความน่าจะเป็นเพื่อจัดการกับปัญหาความไม่แน่นอนของการปรับปรุงค่าความน่าจะเป็นให้มีความสามารถในการตัดสินใจไปในทิศทางที่ถูกต้องได้ดีขึ้น เรียกขั้นตอนที่นำเสนอนี้ว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

จากการปรับปรุงการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับนั้น ผู้วิจัยได้เพิ่มตัวแปรต่าง ๆ เพื่อใช้เก็บข้อมูลค่าความถี่ และค่าความต่อเนื่อง และใช้เป็นข้อมูลประกอบการตัดสินใจในการปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธี ซึ่งตัวแปรที่เพิ่มขึ้นมามีดังนี้

1. *Ufreq* คือ เวกเตอร์ที่ใช้จัดเก็บค่าความถี่เข้าหา 1.0 หรือจำนวนครั้งของการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 ในแต่ละบิตของเวกเตอร์ความน่าจะเป็น
2. *Dfreq* คือ เวกเตอร์ที่ใช้จัดเก็บค่าความถี่เข้าหา 0.0 หรือจำนวนครั้งของการปรับปรุงค่าความน่าจะเป็นเข้าหา 0.0 ในแต่ละบิตของเวกเตอร์ความน่าจะเป็น
3. *Ucon* คือ เวกเตอร์ที่ใช้จัดเก็บค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 ในแต่ละบิตของเวกเตอร์ความน่าจะเป็น ซึ่งถ้าหากการปรับปรุงค่าความน่าจะเป็นเปลี่ยนทิศทางเข้าหา 0.0 ค่าของ *Ucon* ในบิตนั้นจะมีค่าความต่อเนื่องเปลี่ยนเป็น 0 ทันที
4. *Dcon* คือ เวกเตอร์ที่ใช้จัดเก็บค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหา 0.0 ในแต่ละบิตของเวกเตอร์ความน่าจะเป็น ซึ่งถ้าหากการปรับปรุงค่าความน่าจะเป็นเปลี่ยนทิศทางเข้าหา 1.0 ค่าของ *Dcon* ในบิตนั้นจะมีค่าความต่อเนื่องเปลี่ยนเป็น 0 ทันที
5. *Gen* คือ ตัวแปรที่บอกถึงจำนวนของวิวัฒนาการ หรือจำนวนรุ่น

6. p_{size}/n คือ การนำขนาดของประชากร (เมื่อเทียบกับประชากรของขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย) มาหารด้วยจำนวน n (n คือ ส่วนของขนาดประชากร) แล้วได้เป็นอัตราส่วนของขนาดประชากร ซึ่งนำมาใช้ในการตัดสินใจปรับปรุงค่าความน่าจะเป็นด้วยวิธีของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ เมื่อจำนวนรุ่นของการวิวัฒนาการมีค่ามากกว่าอัตราส่วนนี้ รายละเอียดการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่แสดงในภาพที่ 3-2

เมื่อ l คือ ความยาวโครโมโซม

```

for  $i := 1$  to  $l$ 
begin
  if  $winner[i] \neq loser[i]$  then
    if  $winner[i] = 1$  then
       $Ufreq[i] := Ufreq[i] + 1;$ 
       $Ucon[i] := Ucon[i] + 1;$ 
       $Dcon [i] := 0;$ 
      if ( $Ufreq[i] > Dfreq[i]$  AND  $Gen > (psize/4)$ ) then
         $p[i] := p[i] + ((1/psize) + (p[i] * (Ucon[i]/100)));$ 
      else
         $p[i] := p[i] + (1/psize);$ 
      end if
      if ( $p[i] > 1$ )  $p[i] = 1;$ 
    else
       $Dfreq[i] := Dfreq[i] + 1;$ 
       $Dcon [i] := Dcon [i] + 1;$ 
       $Ucon[i] := 0;$ 
      if ( $Dfreq[i] > Ufreq[i]$  AND  $Gen > (psize/4)$ ) then
         $p[i] := p[i] - ((1/psize) + (p[i] * (Dcon [i] / 100)));$ 
      else
         $p[i] := p[i] - (1/psize);$ 
      end if
      if ( $p[i] < 0$ )  $p[i] = 0;$ 
    end if
  end if
end
end

```

ภาพที่ 3-2 การปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

การทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีการจัดเก็บค่าความถี่ และค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าหา 0.0 และ 1.0 เพื่อนำมาใช้ในการตัดสินใจและปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มมากขึ้น โดยนำค่าความถี่มาใช้ในการตัดสินใจ และนำค่าความต่อเนื่องมาใช้ในการปรับปรุงค่าความน่าจะเป็น ซึ่งรายละเอียดของการจัดเก็บข้อมูลค่าความถี่ และค่าความต่อเนื่องมีดังนี้

1 การจัดเก็บข้อมูลค่าความถี่

ในระหว่างการทำงานของขั้นตอนวิธี จะมีการจัดเก็บข้อมูลค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นของแต่ละบิตในเวกเตอร์ความน่าจะเป็น เพื่อนำมาใช้เป็นข้อมูลประกอบการตัดสินใจในการปรับปรุงค่าความน่าจะเป็น โดยค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นนี้สามารถที่จะบอกได้ถึงทิศทางของการปรับปรุงค่าความน่าจะเป็นในบิตนั้นๆ ว่ามีทิศทางไปในทางใดมากกว่ากันระหว่างการปรับปรุงค่าความน่าจะเป็นเข้าหา 1.0 หรือ 0.0 ซึ่งถ้าหากค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 มากกว่าค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 ก็อาจจะหมายความว่าในบิตนี้มีทิศทางของการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 แต่ถ้าหากค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 มากกว่าค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 1.0 ก็อาจจะหมายความว่าในบิตนี้มีทิศทางของการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 และถ้าหากในการปรับปรุงค่าความน่าจะเป็นในครั้งนั้นมีทิศทางที่ตรงกันกับค่าความถี่ของการปรับปรุงค่าความน่าจะเป็นแล้ว ก็แสดงว่ามีความเป็นไปได้ว่าในการปรับปรุงค่าความน่าจะเป็นครั้งนี้มีทิศทางที่ถูกต้อง ตัวอย่างดังตารางที่ 3-2

ตารางที่ 3-2 ตัวอย่างการจัดเก็บค่าความถี่ในการปรับปรุงค่าความน่าจะเป็น

ประชากร		ค่าความถี่	
ประชากรรุ่นที่ 1	โครโมโซมที่ 1 : 0 1 1 1 1 0	ค่าความถี่เข้าใกล้ 1.0 (<i>Ufreq</i>)	0 0 1 1 0 0
	โครโมโซมที่ 2 : 0 1 0 0 1 1	ค่าความถี่เข้าใกล้ 0.0 (<i>Dfreq</i>)	0 0 0 0 0 1
ประชากรรุ่นที่ 2	โครโมโซมที่ 1 : 1 0 1 0 1 1	ค่าความถี่เข้าใกล้ 1.0 (<i>Ufreq</i>)	1 0 2 1 0 0
	โครโมโซมที่ 2 : 0 1 0 1 1 1	ค่าความถี่เข้าใกล้ 0.0 (<i>Dfreq</i>)	0 1 0 1 0 1
ประชากรรุ่นที่ 3	โครโมโซมที่ 1 : 1 1 1 1 0 1	ค่าความถี่เข้าใกล้ 1.0 (<i>Ufreq</i>)	1 0 3 2 0 0
	โครโมโซมที่ 2 : 1 1 0 0 1 1	ค่าความถี่เข้าใกล้ 0.0 (<i>Dfreq</i>)	0 1 0 1 1 1

จากตารางที่ 3-2 มีตัวอย่างของประชากรอยู่ 3 รุ่น โดยแต่ละรุ่นประกอบด้วยประชากร 2 ตัว ซึ่งมีขนาดของโครโมโซม 6 บิต สมมติให้โครโมโซมที่ 1. ของแต่ละรุ่นเป็นประชากรตัวที่ชนะ เมื่อมีการปรับปรุงค่าความน่าจะเป็นในแต่ละรุ่นแล้วจะมีค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในการปรับขึ้นและลงที่เกิดขึ้นในแต่ละบิตของเวกเตอร์ความน่าจะเป็น โดยเรียกว่าค่าความถี่เข้าใกล้ 1.0 และค่าความถี่เข้าใกล้ 0.0 อย่างเช่นในประชากรรุ่นที่ 1 ในตำแหน่งที่ 3 ของโครโมโซมที่ 1 และ 2 เป็นบิตที่มีค่าต่างกัน โดยโครโมโซมตัวที่ 1 มีค่าเป็น 1 ดังนั้นเราจะปรับค่าความถี่เข้าใกล้ 1.0 ในตำแหน่งนั้นเพิ่มขึ้น 1 และในตำแหน่งที่ 6 โครโมโซมตัวที่ 1 มีค่าเป็น 0 ดังนั้นเราจะปรับค่าความถี่เข้าใกล้ 0.0 ในตำแหน่งนั้นเพิ่มขึ้น 1

2 การจัดเก็บข้อมูลค่าความต่อเนื่อง

เป็นการจัดเก็บข้อมูลค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นของแต่ละบิตในเวกเตอร์ความน่าจะเป็น เพื่อนำมาใช้ในการปรับปรุงค่าความน่าจะเป็นในแต่ละบิตของเวกเตอร์ความน่าจะเป็น โดยค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นนี้สามารถที่จะบอกได้ถึงความขัดแย้งที่เกิดขึ้นในการปรับปรุงค่าความน่าจะเป็นในแต่ละบิต ซึ่งถ้าหากในบิตใดมีค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นน้อย แสดงว่าในบิตนั้นก็มีความขัดแย้งในการปรับปรุงค่าความน่าจะเป็นมาก และจะทำการปรับค่าความต่อเนื่องให้เป็นศูนย์ เมื่อมีการปรับปรุงไปในทิศทางตรงกันข้าม แต่ถ้าหากในบิตใดมีค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นมาก แสดงว่าในบิตนั้นก็มีความขัดแย้งในการปรับปรุงค่าความน่าจะเป็นน้อย นั่นหมายความว่า

เราสามารถที่จะมั่นใจได้ว่าการปรับปรุงค่าความน่าจะเป็นในบิตนั้นมีการปรับปรุงไปในทิศทางที่ชัดเจนตัวอย่าง ดังตารางที่ 3-3

ตารางที่ 3-3 ตัวอย่างการจัดเก็บค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็น

ประชากร		ค่าความต่อเนื่อง
ประชากรรุ่นที่ 1	โครโมโซมที่ 1 : 0 1 1 1 1 0	ค่าความต่อเนื่องเข้าใกล้ 1.0 (<i>Ucon</i>)
	โครโมโซมที่ 2 : 0 1 0 0 1 1	ค่าความต่อเนื่องเข้าใกล้ 0.0 (<i>Dcon</i>)
ประชากรรุ่นที่ 2	โครโมโซมที่ 1 : 1 0 1 0 1 1	ค่าความต่อเนื่องเข้าใกล้ 1.0 (<i>Ucon</i>)
	โครโมโซมที่ 2 : 0 1 0 1 1 1	ค่าความต่อเนื่องเข้าใกล้ 0.0 (<i>Dcon</i>)
ประชากรรุ่นที่ 3	โครโมโซมที่ 1 : 1 1 1 1 0 1	ค่าความต่อเนื่องเข้าใกล้ 1.0 (<i>Ucon</i>)
	โครโมโซมที่ 2 : 1 1 0 0 1 1	ค่าความต่อเนื่องเข้าใกล้ 0.0 (<i>Dcon</i>)

จากตารางที่ 3-3 มีตัวอย่างของประชากรอยู่ 3 รุ่น โดยแต่ละรุ่นประกอบด้วยประชากร 2 ตัว ซึ่งมีขนาดของโครโมโซม 6 บิต สมมุติให้โครโมโซมที่ 1. ของแต่ละรุ่นเป็นประชากรตัวที่ชนะ เมื่อมีการปรับปรุงค่าความน่าจะเป็นในแต่ละรุ่นแล้วจะมีค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นเข้าใกล้ 1.0 และ 0.0 เกิดขึ้นในแต่ละบิตของเวกเตอร์ความน่าจะเป็น โดยเรียกว่าค่าความต่อเนื่องเข้าใกล้ 1.0 และค่าความต่อเนื่องเข้าใกล้ 0.0 ตัวอย่างเช่นในประชากรรุ่นที่ 2 ในตำแหน่งที่ 3 ของโครโมโซมที่ 1 และ 2 เป็นบิตที่มีค่าต่างกัน โดยโครโมโซมตัวที่ 1 มีค่าเป็น 1 ดังนั้นเราจะปรับค่าความต่อเนื่องเข้าใกล้ 1.0 ในตำแหน่งนั้นเพิ่มขึ้น 1 และในตำแหน่งที่ 4 โครโมโซมตัวที่ 1 มีค่าเป็น 0 ดังนั้นเราจะปรับค่าความต่อเนื่องเข้าใกล้ 0.0 ในตำแหน่งนั้นเพิ่มขึ้น 1 แต่ค่าความต่อเนื่องเข้าใกล้ 1.0 ในตำแหน่งนั้นจะต้องทำการปรับกลับมาเป็น 0 ด้วย

3 การปรับปรุงค่าความน่าจะเป็น

จากการศึกษาทดลองนำค่าความถี่ และค่าความต่อเนื่องมาใช้ในการปรับปรุงขั้นตอนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับและได้เป็นขั้นตอนวิธีใหม่ที่เรียกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

ด้วยค่าความถี่นั้น ผู้วิจัยได้นำค่าความถี่ และค่าความต่อเนื่องมาใช้ในขั้นตอนการปรับปรุงค่าความน่าจะเป็น ซึ่งการปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีรายละเอียดดังนี้

1. ถ้าผู้ชนะมีค่าเป็น 1

ในกรณีที่ผู้ชนะมีค่าเป็น 1 นั้น ถ้าหากค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 1 มีค่ามากกว่าค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 0 หมายความว่า การปรับปรุงค่าความน่าจะเป็นในบิตนี้มีความน่าเชื่อถือที่ทำให้เชื่อถือได้ว่าเป็นไปในทิศทางที่นำไปสู่คำตอบที่ดีขึ้น และเมื่อจำนวนของการวิวัฒนาการมีค่ามากกว่า $psize/n$ แล้วนั้น ขั้นตอนวิธีจะทำการปรับปรุงค่าความน่าจะเป็นโดยใช้ค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นของบิตนี้ แต่ถ้าหากค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 1 น้อยกว่าค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 0 หรือจำนวนของการวิวัฒนาการน้อยกว่า $psize/n$ ขั้นตอนวิธีนี้จะทำการปรับปรุงค่าความน่าจะเป็นด้วยการปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับแบบดั้งเดิม

2. ถ้าผู้ชนะมีค่าเป็น 0

ในกรณีที่ผู้ชนะมีค่าเป็น 0 นั้น ถ้าหากค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 0 มีค่ามากกว่าค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 1 หมายความว่า การปรับปรุงค่าความน่าจะเป็นในบิตนี้มีความน่าเชื่อถือที่ทำให้เชื่อถือได้ว่าเป็นไปในทิศทางที่ถูกต้อง และเมื่อจำนวนของการวิวัฒนาการมีค่ามากกว่า $psize/n$ แล้วนั้น ขั้นตอนวิธีจะทำการปรับปรุงค่าความน่าจะเป็นด้วยค่าความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นของบิตนี้ แต่ถ้าหากค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 0 น้อยกว่าค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นในทิศทางเข้าใกล้ 1 หรือจำนวนของการวิวัฒนาการน้อยกว่า $psize/n$ ขั้นตอนวิธีนี้จะทำการปรับปรุงค่าความน่าจะเป็นด้วยวิธีการของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับแบบดั้งเดิม

พารามิเตอร์ $psize/n$

จากที่ผู้วิจัยได้นำเสนอการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ จะเห็นได้ว่าผู้วิจัยนำพารามิเตอร์ $psize/n$ มาเป็นเงื่อนไขหนึ่งในการตัดสินใจปรับปรุงค่าความน่าจะเป็น ซึ่งพารามิเตอร์ $psize/n$ คือขนาดของประชากรหารด้วยจำนวน n (เมื่อ n คือ ส่วนของขนาดประชากร) แล้วได้เป็นอัตราส่วนของขนาดประชากร นำมาใช้ในการตัดสินใจปรับปรุงค่าความน่าจะเป็นเมื่อจำนวนของวิวัฒนาการมีค่ามากกว่าอัตราส่วนของขนาดประชากรนี้ เนื่องจากการทำงานรอบแรก ๆ ของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับนั้นยังไม่มีทิศทางที่แน่นอนในการปรับปรุงค่าความน่าจะเป็น และจากการศึกษาของผู้วิจัยพบว่าขั้นตอนนี้จะเริ่มมีทิศทางที่ชัดเจนในการปรับปรุงค่าความน่าจะเป็นเมื่อผ่านวิวัฒนาการไปแล้วระยะหนึ่ง ดังนั้นผู้วิจัยจึงได้ทำการศึกษาค่าพารามิเตอร์ $psize/n$ เพื่อศึกษาถึงอัตราส่วนที่เหมาะสมที่จะนำมาใช้ในขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

ในการศึกษาผู้วิจัยได้ทำการทดสอบค่าพารามิเตอร์ $psize/n$ กับปัญหาจำนวนบิตหนึ่งมากที่สุด, ปัญหาเรนคอมแม็ก, ปัญหารอยัล โรด และปัญหาคับคัก ที่ขนาดของ tournament 2, 4, 8 และแทนค่า n ด้วยค่า 2, 3, 4 และ 5 หรือ $psize/2$, $psize/3$, $psize/4$ และ $psize/5$ เพื่อทดสอบประสิทธิภาพการหาคำตอบ และจำนวนครั้งในการประเมินค่าความเหมาะสมของแต่ละพารามิเตอร์ แล้วนำผลการทดลองที่ได้มาคำนวณเพื่อเปรียบเทียบอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินของแต่ละพารามิเตอร์ ซึ่งสูตรที่ใช้ในการคำนวณหาค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินนั้น แสดงดังสมการที่ 3.1

$$\text{อัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมิน} = \frac{\text{ค่าความเหมาะสมที่ดีที่สุดที่ขั้นตอนวิธีหามาได้}}{\text{จำนวนครั้งในการประเมินค่าความเหมาะสม}} \quad (3.1)$$

จากสมการที่ 3.1 เป็นสมการที่ใช้เพื่อคำนวณหาค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินในการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ โดยการนำค่าความเหมาะสมที่ดีที่สุดที่ขั้นตอนวิธีหามาได้มาหารด้วยจำนวนครั้งในการประเมินค่าความเหมาะสม ซึ่งค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินที่มากนั้นจะแสดงถึงประสิทธิภาพในการหาคำตอบที่สูง

อัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินนำมาใช้เพื่อวัดประสิทธิภาพของแต่ละพารามิเตอร์ที่ทำการทดสอบ ซึ่งเป็นอัตราส่วนระหว่างค่าความเหมาะสมที่ดีที่สุดที่ขั้นตอนวิธีหามาได้ กับจำนวนครั้งในการประเมินค่าความเหมาะสม โดยค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินที่สูงแสดงถึงประสิทธิภาพในการหาคำตอบที่สูง ซึ่งจากตารางที่ 3-4 แสดงให้เห็นอัตราส่วนค่าความเหมาะสมต่อ

จำนวนครั้งการประเมินจากทุกปัญหา และอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินโดยเฉลี่ย โดย $p_{size}/4$ มีค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินเฉลี่ยที่สูงใน tournament size 4 และ 8 ซึ่งใน tournament size 4 มีค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินเฉลี่ย 0.0340 และใน tournament size 8 มีค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินเฉลี่ย 0.0418 และในขนาดของ tournament size 2 นั้น $p_{size}/3$ มีค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินที่สูงกว่าพารามิเตอร์อื่น โดยมีค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินเฉลี่ย 0.0249 จากการวัดอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินโดยรวมนั้น ค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินที่สูงที่สุดมาจาก $p_{size}/4$ ซึ่งค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินโดยเฉลี่ยจากทุกปัญหา และขนาดของ tournament นั้น $p_{size}/4$ มีค่าเท่ากับ 0.0330 ดังตารางที่ 3-5

ตารางที่ 3-4 ผลการวัดอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมิน

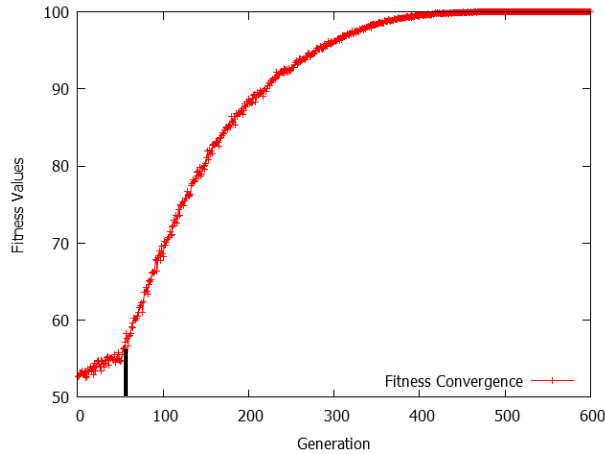
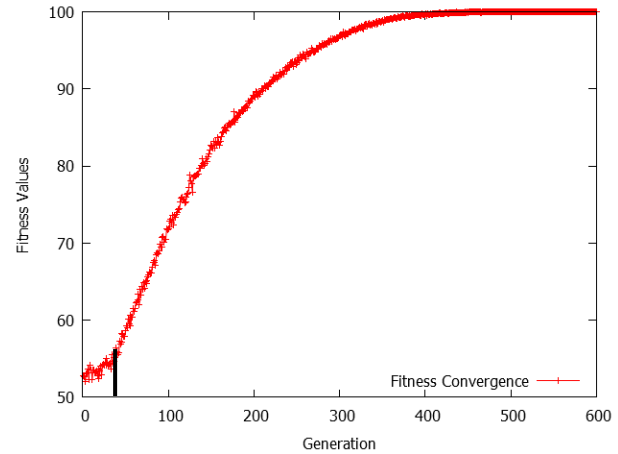
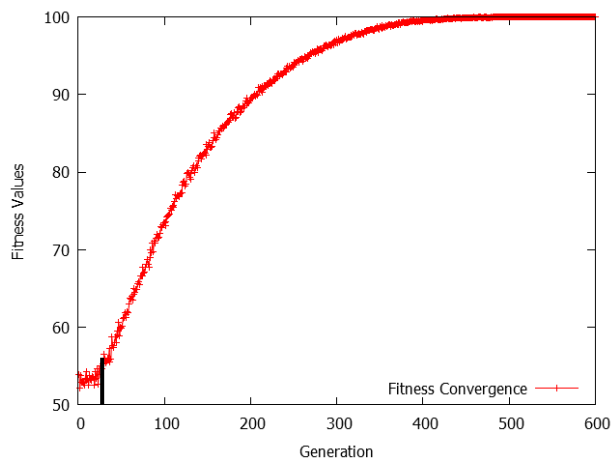
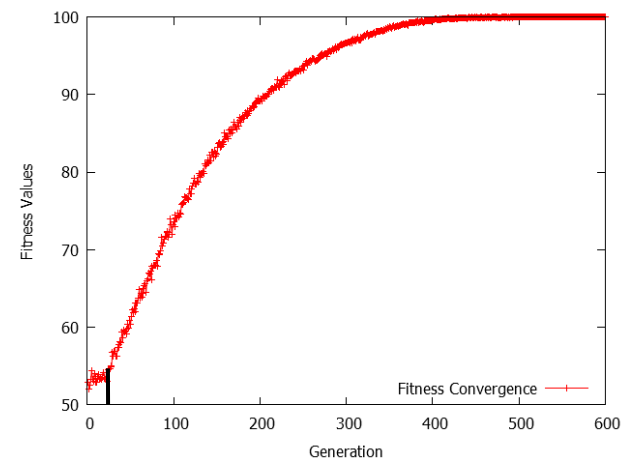
Problem		อัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมิน				ค่าเฉลี่ย
		One-Max	Random Max	Royal Road	Trap	
Tournament t Size 2	$p_{size}/2$	0.0557	0.0333	0.0051	0.0004	0.0236
	$p_{size}/3$	0.0575	0.0363	0.0053	0.0005	0.0249
	$p_{size}/4$	0.0528	0.0353	0.0043	0.0004	0.0232
	$p_{size}/5$	0.0547	0.0345	0.0049	0.0004	0.0237
Tournament t Size 4	$p_{size}/2$	0.0764	0.0429	0.0095	0.0008	0.0324
	$p_{size}/3$	0.0762	0.0423	0.0102	0.0009	0.0324
	$p_{size}/4$	0.0826	0.0419	0.0107	0.0010	0.0340
	$p_{size}/5$	0.0790	0.0409	0.0119	0.0009	0.0332
Tournament t Size 8	$p_{size}/2$	0.0850	0.0403	0.0148	0.0007	0.0352
	$p_{size}/3$	0.0784	0.0425	0.0146	0.0009	0.0341
	$p_{size}/4$	0.1058	0.0472	0.0133	0.0011	0.0418
	$p_{size}/5$	0.1014	0.0456	0.0139	0.0011	0.0405

ตารางที่ 3-5 ค่าอัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินโดยเฉลี่ยจากทุกปัญหา และขนาดของ tournament

อัตราส่วนค่าความเหมาะสมต่อจำนวนครั้งการประเมินโดยเฉลี่ย	
psize/2	0.0304
psize/3	0.0305
psize/4	0.0330
psize/5	0.0325

ภาพที่ 3-3 เป็นกราฟที่แสดงถึงการลู่เข้าสู่คำตอบของค่าความเหมาะสมที่เกิดขึ้นตั้งแต่ประชากรรุ่นแรกจนถึงประชากรรุ่นสุดท้ายเมื่อทดสอบกับค่าพารามิเตอร์ $psize/2$, $psize/3$, $psize/4$ และ $psize/5$ ตามลำดับ ซึ่งกราฟที่แสดงเป็นผลการทดลองกับปัญหาจำนวนบิตหนึ่งมากที่สุดที่ประชากรมีขนาด 100 บิต และขนาดของ tournament เท่ากับ 2 โดยแกนนอนคือรุ่นของวิวัฒนาการ แกนตั้งคือค่าความเหมาะสม และเส้นสีดำคือตำแหน่งที่รุ่นของวิวัฒนาการมีค่าเท่ากับ $psize/n$

ผลการทดลองแสดงให้เห็นว่าเมื่อรุ่นของวิวัฒนาการมีค่าเท่ากับ $psize/n$ ขั้นตอนวิธีที่นำเสนอจะเริ่มมีค่าความเหมาะสมลู่เข้าสู่คำตอบได้อย่างรวดเร็ว ซึ่งเกิดจากการที่เมื่อรุ่นของวิวัฒนาการมีค่าเท่ากับ $psize/n$ แล้ว ขั้นตอนวิธีที่นำเสนอมีการปรับปรุงเวกเตอร์ความน่าจะเป็นด้วยค่าความน่าจะเป็นที่มากขึ้น และจากผลการทดลองเมื่อ n มีค่ามากขึ้นขั้นตอนวิธีที่นำเสนอก็ยังมีการเริ่มต้นลู่เข้าสู่คำตอบด้วยรุ่นของวิวัฒนาการที่น้อยลงเรื่อย ๆ ดังภาพที่ 3-3

(a) $psize/2$ (b) $psize/3$ (c) $psize/4$ (d) $psize/5$

ภาพที่ 3-3 การลู่เข้าหาคำตอบของการทดลองกับพารามิเตอร์ $psize/2$, $psize/3$, $psize/4$ และ $psize/5$ จากปัญหาบิตหนึ่งมากที่สุด โดยใช้ขนาด tournament 2

จากภาพที่ 3-3 แสดงให้เห็นว่ายิ่งค่า n มากขึ้นจำนวนรุ่นของวิวัฒนาการที่เริ่มลู่เข้าหาคำตอบก็ยิ่งน้อยลง ซึ่งจากภาพจะเห็นว่า $psize/5$ มีการลู่เข้าหาคำตอบที่เร็วที่สุด ซึ่งเร็วกว่า $psize/4$ เล็กน้อย แต่เนื่องจากผลการวัดประสิทธิภาพได้แสดงให้เห็นว่า $psize/4$ มีค่าประสิทธิภาพโดยเฉลี่ยที่สูงกว่า $psize/5$ ดังนั้นในการทดลองขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่นั้น ผู้วิจัยจึงเลือกใช้ $psize/4$ ในการดำเนินการทดลอง ซึ่งจะอธิบายในบทที่ 4

บทที่ 4

ผลการทดลอง

ในบทนี้เป็นการนำเสนอวิธีการทดลอง และผลการทดลองของขั้นตอนวิธีเชิงพันธุกรรมแบบ กระชับด้วยค่าความถี่ (fb-cGA) และวิเคราะห์การทำงานของขั้นตอนวิธี โดยในการทดลองผู้วิจัยได้นำ ขั้นตอนวิธีที่นำเสนอมาทดสอบกับปัญหาต่าง ๆ ได้แก่

1. ปัญหาจำนวนบิตหนึ่งมากที่สุด (One-Max Problem)
2. ปัญหาเรนดอม แม็กซ์ (Random Max Problem)
3. ปัญหารอยัล โรด (Royal Road Problem)
4. ปัญหาที่กับดัก (Trap Problem)

โดยผลลัพธ์จากการทดสอบกับแต่ละปัญหาจะถูกนำมาเปรียบเทียบกับขั้นตอนวิธีที่มีผู้นำนเสนอมาก่อนหน้า ขั้นตอนวิธีที่ได้นำมาเปรียบเทียบ ได้แก่

1. ขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย (sGA)
2. ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ (cGA)
3. ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย (mcGA)
4. ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร (pe-cGA)
5. ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว (ne-cGA)

ขั้นตอนการทดลอง

1. ทดลองขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่กับปัญหาจำนวนบิตหนึ่งมากที่สุด, ปัญหาเรนดอมแม็กซ์, ปัญหารอยัล โรด และปัญหาที่กับดัก โดยผลการทดลองที่ได้เป็นค่าเฉลี่ยจากการทดสอบโปรแกรมทั้งหมด 50 รอบการทำงาน และใช้พารามิเตอร์ $p_{size}/4$

2. เปรียบเทียบผลการทดลองกับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับโดยใช้ตัวเก่งแบบถาวร และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับโดยใช้ตัวเก่งแบบชั่วคราว โดยทำการทดสอบขั้นตอนวิธีเชิงพันธุกรรมเหล่านี้กับปัญหาจำนวนบิตหนึ่งมากที่สุด, ปัญหาแรนคอมเม็ก, ปัญหารอยัล โรด และปัญหาภัยคุก เช่นเดียวกับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ แล้วนำผลการทดลองเฉลี่ยจากการทดสอบโปรแกรม 50 ครั้งมาเปรียบเทียบผลการทดลอง
3. วิเคราะห์การทำงานของขั้นตอนวิธีจากค่าความน่าจะเป็น และค่าความเหมาะสม โดยการนำค่าความน่าจะเป็น และค่าความเหมาะสมตั้งแต่ประชากรรุ่นแรกจนถึงประชากรรุ่นสุดท้ายมาจัดทำกราฟของค่าความน่าจะเป็น และกราฟของค่าความเหมาะสม เพื่อแสดงถึงการทำงานที่เกิดขึ้น

พารามิเตอร์ที่ใช้ในการทดลอง

พารามิเตอร์ต่าง ๆ ที่ใช้ในการทดลองประกอบด้วยพารามิเตอร์ต่าง ๆ ดังนี้

1. ความยาวโครโมโซม
2. ขนาดประชากร
3. ขนาด tournament

ในการกำหนดค่าของพารามิเตอร์ต่าง ๆ ในแต่ละปัญหา บางพารามิเตอร์จะมีความแตกต่างกันตามรูปแบบของปัญหานั้น ๆ ซึ่งค่าพารามิเตอร์ที่ใช้กับปัญหาต่าง ๆ แสดงดังตารางที่ 4-1

ตารางที่ 4-1 พารามิเตอร์ที่ใช้กับปัญหาต่าง ๆ

ปัญหา	ความยาวโครโมโซม	ขนาดประชากร	ขนาด tournament
ปัญหาจำนวนบิตหนึ่งมากที่สุด	100 บิต	2, 4, 6, 8, 10 เพิ่มขึ้นทีละ 2 จนถึง 100	2
ปัญหาเรนดอม แม็กซ์	100 บิต	2, 4, 6, 8, 10 เพิ่มขึ้นทีละ 2 จนถึง 100	2
ปัญหารอยัล ไรด	64 บิต	2, 4, 6, 8, 10 เพิ่มขึ้นทีละ 2 จนถึง 100	2 4 และ 8
ปัญหากลบดัก	30 บิต (กลบดักความยาว 3 บิตจำนวน 10 ชุด)	8, 500, 1000, 1500, 2000, 2500 และ 3000	2 4 และ 8

ผลการทดลอง

ในส่วนนี้ผู้วิจัยนำเสนอถึงผลการทดลองของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ โดยจะเสนอผลการทดลอง และการวิเคราะห์ใน 3 ด้าน ได้แก่

1. คุณภาพคำตอบ (Solution Quality)
2. จำนวนครั้งในการประเมินค่าความเหมาะสม (Number of Fitness Evaluations)
3. เวกเตอร์ความน่าจะเป็น (Probability Vector)

ผลการทดลองกับปัญหาจำนวนบิตหนึ่งมากที่สุด

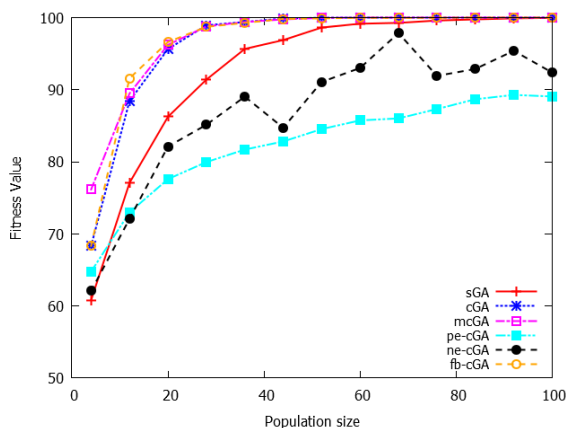
ในการทดลองแรกผู้วิจัยทำการทดลองขั้นตอนวิธีกับปัญหาจำนวนบิตหนึ่งมากที่สุด ซึ่งจากภาพที่ 4-1 เป็นผลการทดลองกับปัญหาจำนวนบิตหนึ่งมากที่สุด ความยาวโครโมโซม 100 บิต ขนาดของ tournament เท่ากับ 2 และเป็นผลการทดลองจาก 50 รอบการทำงานโดยเฉลี่ย โดยภาพที่ 4-1ก แสดงถึงค่าความเหมาะสม เมื่อทดสอบกับขนาดความยาวโครโมโซมต่าง ๆ และภาพที่ 4-1ข แสดงถึงจำนวน

ครั้งในการประเมินค่าความเหมาะสม ซึ่งผลการทดลองทั้งหมดผู้วิจัยได้ทำการทดลอง และจัดเก็บข้อมูลเอง

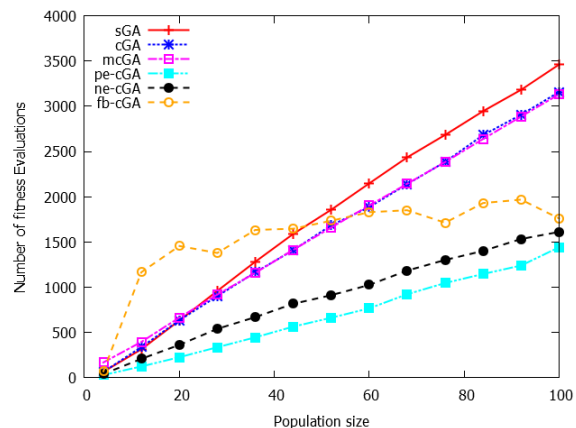
จากภาพที่ 4-1(ก) ผลการทดลองแสดงให้เห็นถึงค่าความเหมาะสม โดยขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่สามารถหาคำตอบที่ดีที่สุดได้เมื่อขนาดประชากรมากกว่า 40 ขึ้นไป ซึ่งสามารถเทียบได้กับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย

จากภาพที่ 4-1(ข) แสดงให้เห็นว่าในขนาดประชากรที่ต่ำกว่า 40 นั้น ขั้นตอนวิธีมีจำนวนครั้งในการประเมินค่าความเหมาะสมที่มากกว่าขั้นตอนวิธีอื่น แต่เมื่อขนาดของประชากรเพิ่มมากขึ้น ขั้นตอนวิธีที่นำเสนอก็มีจำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบธรรมดาอย่างง่าย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย

ถึงแม้ว่าภาพที่ 4-1(ก) จะแสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่ดีกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราวก็ตาม แต่จากภาพที่ 4-1(ข) แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีจำนวนครั้งในการประเมินค่าความเหมาะสมที่มากกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว



(ก) ค่าความเหมาะสม



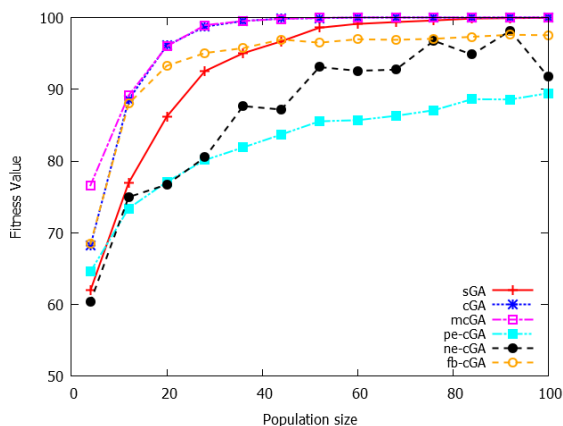
(ข) จำนวนครั้งในการประเมินค่าความเหมาะสม

ภาพที่ 4-1 ผลการทดลองจากปัญหาจำนวนบิตหนึ่งมากที่สุด

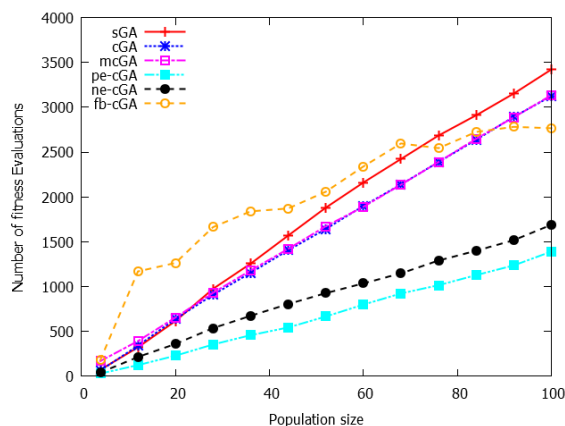
ผลการทดลองกับปัญหาแรนคอมแม็ก

ภาพที่ 4-2 แสดงผลการทดลองขั้นตอนวิธีที่นำเสนอกับปัญหาแรนคอมแม็ก ซึ่งผลการทดลองทั้งหมดผู้วิจัยได้ทำการทดลอง และจัดเก็บข้อมูลเอง โดยภาพที่ 4-2(ก) คือค่าความเหมาะสม และภาพที่ 4-2(ข) คือจำนวนครั้งในการประเมินค่าความเหมาะสม ซึ่งจากภาพที่ 4-2(ก) ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่สามารถเทียบได้กับขั้นตอนวิธีพันธุกรรมแบบกระชับโดยใช้ตัวเก่งแบบชั่วคราว และมีค่าความเหมาะสมที่มากกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับโดยใช้ตัวเก่งแบบถาวร

จากภาพที่ 4-2(ข) ในด้านจำนวนครั้งในการประเมินค่าความเหมาะสม ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ต้องการจำนวนครั้งในการประเมินค่าความเหมาะสมมากในขนาดของประชากรที่มีขนาดเล็ก แต่เมื่อประชากรมีขนาดที่เพิ่มขึ้นจำนวนครั้งในการประเมินค่าความเหมาะสมก็มีแนวโน้มที่ลดน้อยลง และสามารถเทียบได้กับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย



(ก) ค่าความเหมาะสม



(ข) จำนวนครั้งในการประเมินค่าความเหมาะสม

ภาพที่ 4-2 ผลการทดลองจากปัญหาเรนคอมแม็ก

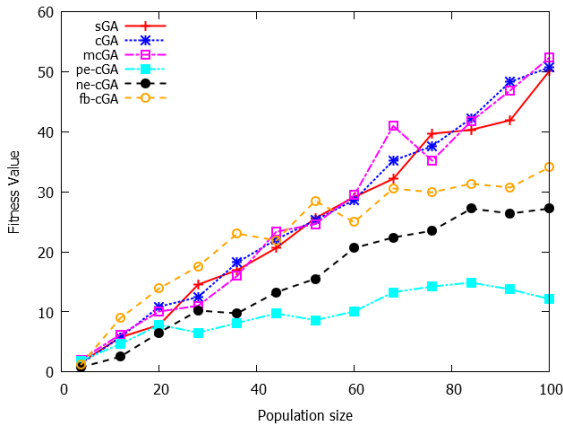
ผลการทดลองกับปัญหาฮารอลโรด

ภาพที่ 4-3 แสดงผลการทดลองของขั้นตอนวิธีต่าง ๆ กับปัญหาฮารอลโรดที่ขนาดของ tournament เท่ากับ 2, 4 และ 8 ความยาวโครโมโซม 64 บิต และขนาดประชากร 4 – 100 โดยภาพที่ 4-3(ก) , 4-3(ค) และ 4-3(จ) แสดงถึงค่าความเหมาะสม และภาพที่ 4-3(ข) , 4-3(ง) และ 4-3(ฉ) คือ จำนวนครั้งในการประเมินค่าความเหมาะสม ผลการทดลองทั้งหมดผู้วิจัยได้ทำการทดลอง และจัดเก็บข้อมูลเอง ซึ่งผลการทดลองที่ได้มีดังนี้

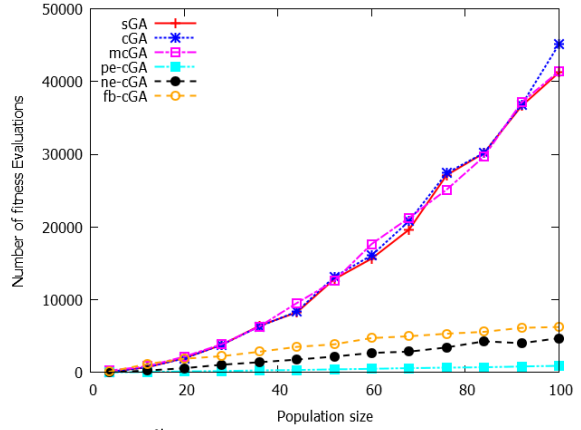
1. ขั้นตอนวิธีเชิงพันธุกรรมด้วยค่าความถี่มีค่าความเหมาะสมที่สามารถเทียบได้กับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ยแต่มีจำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยกว่าหลายเท่า
2. เมื่อเปรียบเทียบกับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว ที่ขนาดของ tournament เท่ากับ 2 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่ดีกว่า แต่ต้องการจำนวนในการประเมินค่าความเหมาะสมที่มากกว่า แต่อย่างไรก็ตามที่ขนาดของ tournament เท่ากับ 4 และ 8 ขั้นตอนวิธีเชิงพันธุกรรมแบบ

กระชับด้วยค่าความถี่มีค่าความเหมาะสมที่เทียบได้กับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว แต่ต้องการจำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยกว่า

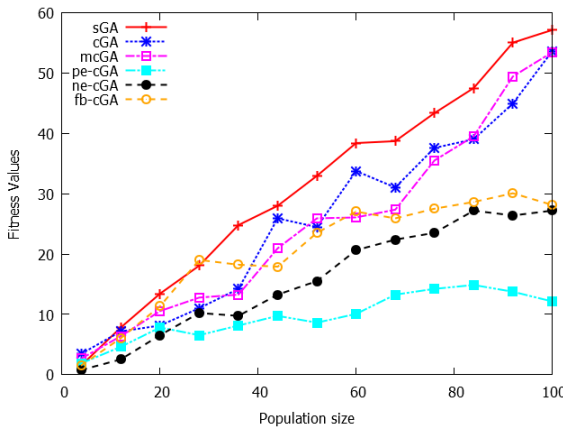
3. ถึงแม้ว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งถาวรจะต้องการจำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ก็ตาม แต่ในด้านของค่าความเหมาะสมขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่ดีกว่า



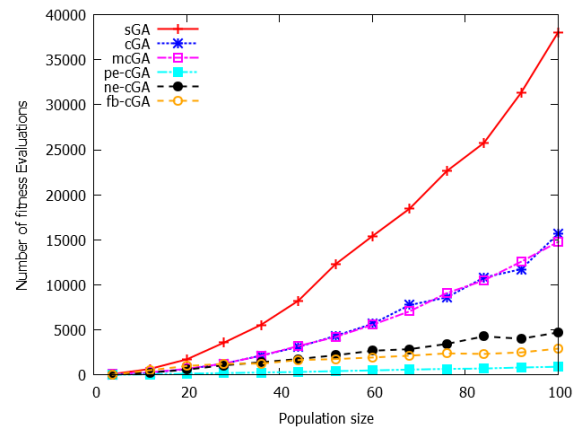
(ก) ค่าความเหมาะสม (Tournament size 2)



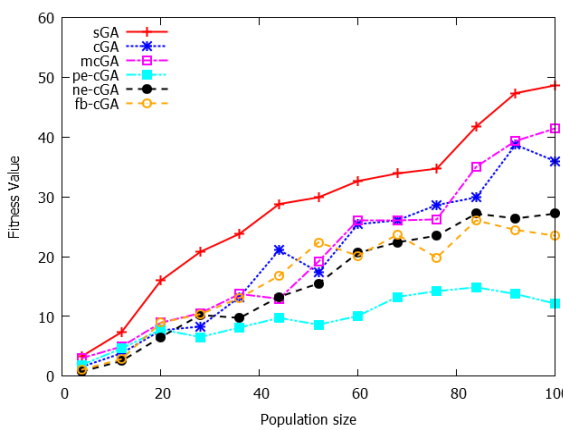
(ข) จำนวนครั้งในการประเมินค่าความเหมาะสม



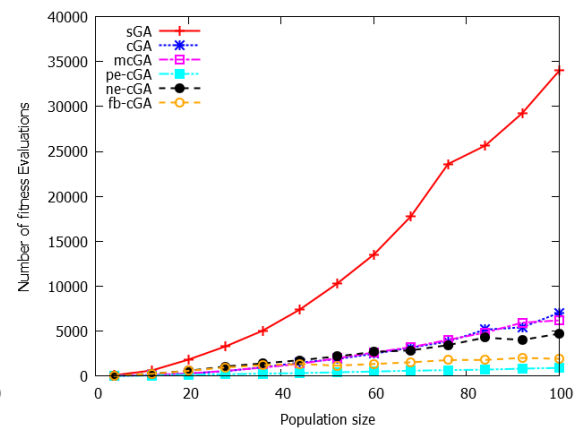
(ค) ค่าความเหมาะสม (Tournament size 4)



(ด) จำนวนครั้งในการประเมินค่าความเหมาะสม



(จ) ค่าความเหมาะสม (Tournament size 8)



(ฉ) จำนวนครั้งในการประเมินค่าความเหมาะสม

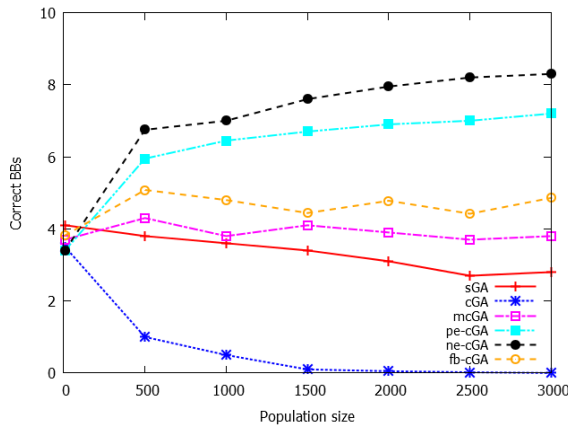
ภาพที่ 4-3 ผลการทดลองจากปัญหาการยัลโรค

ผลการทดลองกับปัญหาภัยคุกคาม

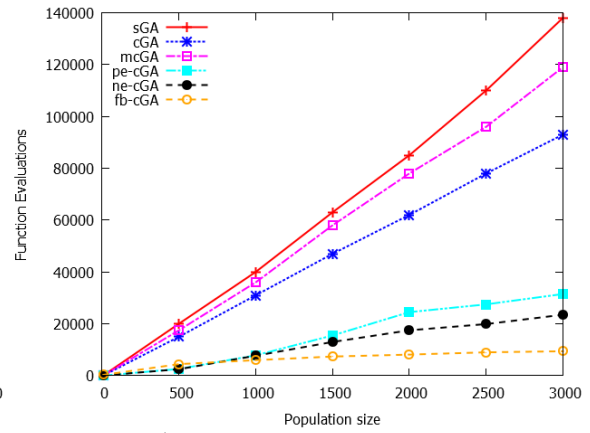
ภาพที่ 4-4 แสดงผลการทดลองจากปัญหาภัยคุกคามขนาด 3x10 บิต ที่ขนาดของ tournament เท่ากับ 2, 4 และ 8 และขนาดของประชากรเท่ากับ 8, 500, 1000, 1500, 2000, 2500 และ 3000 โดยภาพที่ 4-4(ก), 4-4(ค) และ 4-4(จ) คือค่าความเหมาะสม แนวตั้งคือ คุณภาพของคำตอบ (Correct BBs) ซึ่งคือ จำนวนชุดของปัญหาภัยคุกคามที่ทุกบิตในชุดนั้นเป็น 1 ทั้งหมด และแนวนอนคือ ขนาดของประชากร (Population Size) และภาพที่ 4-4(ข), 4-4(ง) และ 4-4(ฉ) คือจำนวนครั้งในการประเมินค่าความเหมาะสม แนวตั้งคือ จำนวนครั้งในการประเมินค่าความเหมาะสม (Fitness Evaluations) และแนวนอนคือ ขนาดของประชากร โดยผลการทดลองจากขั้นตอนวิธีอื่น ๆ ที่มีผู้นำเสนอมาก่อนหน้านี้ได้นำมาจากบทความที่ขั้นตอนวิธีนั้นตีพิมพ์ ซึ่งผลการทดลองที่ได้มีดังนี้

1. ที่ขนาดของ tournament เท่ากับ 2 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่ดีกว่า ขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย และมีค่าความเหมาะสมที่ต่ำกว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว
2. ที่ขนาดของ tournament เท่ากับ 4 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่ดีกว่า ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และเทียบได้กับขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร แต่มีค่าความเหมาะสมที่ต่ำกว่าขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว
3. ที่ขนาดของ tournament เท่ากับ 8 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีค่าความเหมาะสมที่ดีกว่า ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว แต่มีค่าความเหมาะสมที่ต่ำกว่าขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายและขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย

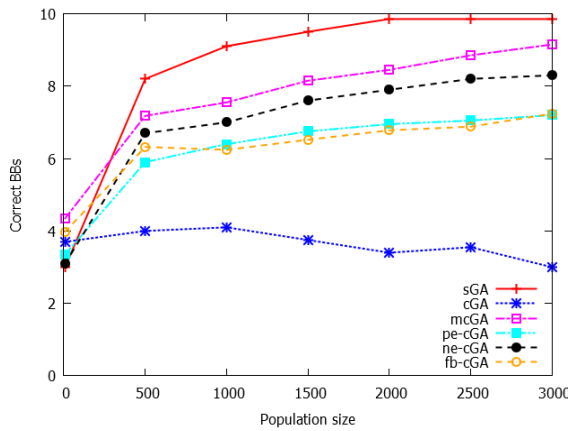
4. ในด้านของจำนวนครั้งในการประเมินค่าความเหมาะสม ขั้นตอนวิธีเชิงพันธุกรรมแบบ กระชับด้วยค่าความถี่มีจำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยที่สุด และมี จำนวนที่น้อยกว่าขั้นตอนวิธีอื่นหลายเท่าในทุกขนาดของ tournament ซึ่งแสดงให้เห็น ถึงประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ที่ซึ่งนำเสนอถึง การใช้ระยะเวลาในการหาคำตอบที่น้อย



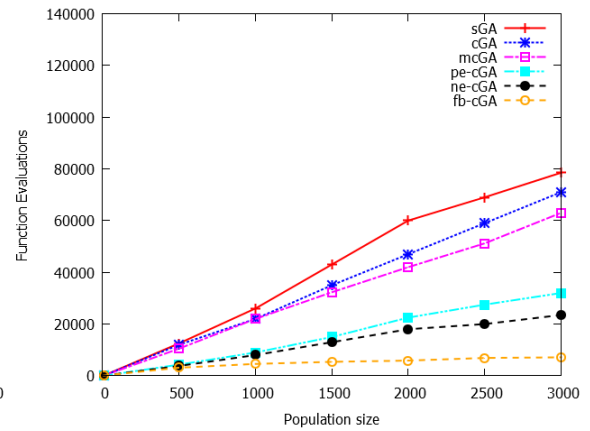
(ก) ค่าความเหมาะสม (Tournament size 2)



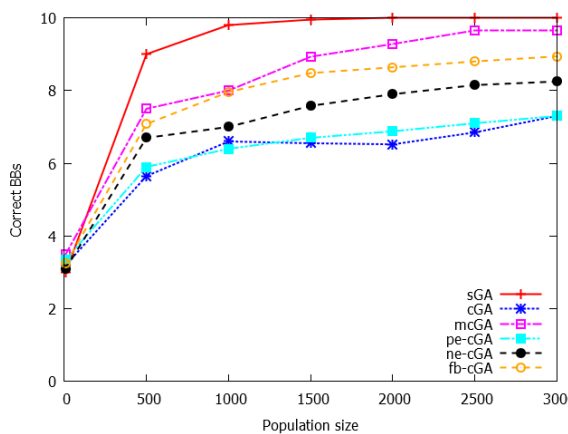
(ข) จำนวนครั้งในการประเมินค่าความเหมาะสม



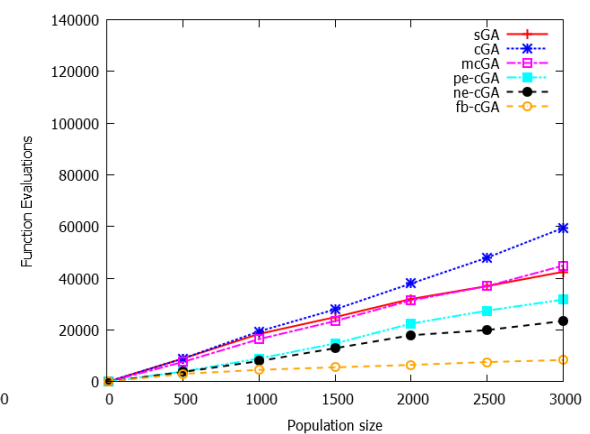
(ค) ค่าความเหมาะสม (Tournament size 4)



(ด) จำนวนครั้งในการประเมินค่าความเหมาะสม



(จ) ค่าความเหมาะสม (Tournament size 8)



(ฉ) จำนวนครั้งในการประเมินค่าความเหมาะสม

ภาพที่ 4-4 ผลการทดลองจากปัญหาหักับคัก

การวิเคราะห์เวกเตอร์ความน่าจะเป็น

ภาพที่ 4-5 แสดงถึงการเปลี่ยนแปลงค่าความน่าจะเป็นในเวกเตอร์ความน่าจะเป็นในแต่ละรอบการทำงาน(รุ่น) และค่าความเหมาะสมจากขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบถาวร, ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราว และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ สำหรับปัญหาชนิดหนึ่งมากที่สุด ที่ขนาดของ tournament เท่ากับ 2 และขนาดประชากรเท่ากับ 100 โดยผลการทดลองที่ได้เกิดจากค่าเฉลี่ยจาก 50 รอบ

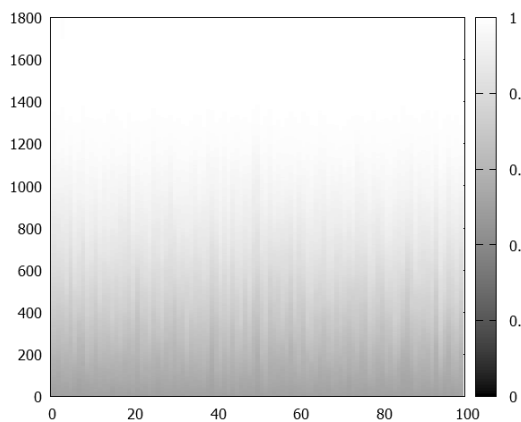
กราฟทางด้านซ้ายแสดงถึงค่าความน่าจะเป็นตั้งแต่ประชากรรุ่นแรกจนถึงประชากรรุ่นสุดท้าย ซึ่งอยู่ในรูปแบบของภาพขาวดำ โดยค่า 0 แทนด้วยโทนสีดำ และค่า 1 แทนด้วยโทนสีขาว แกนนอนคือค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็น และแกนตั้งคือจำนวนของรุ่นของประชากร

กราฟทางด้านขวาแสดงถึงค่าความเหมาะสมที่เกิดขึ้นตั้งแต่ประชากรรุ่นแรกจนถึงประชากรรุ่นสุดท้าย โดยแกนนอนคือรุ่นของประชากร และแกนตั้งคือค่าความเหมาะสม

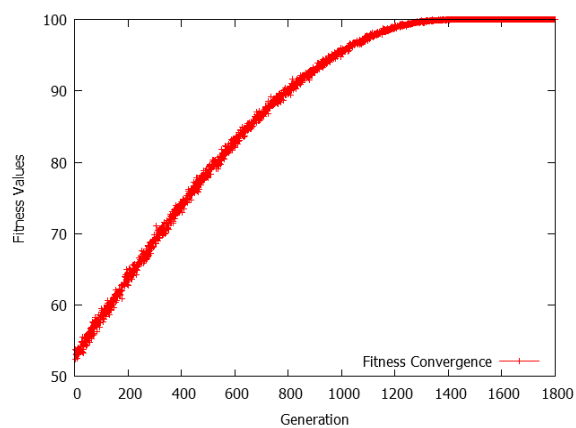
จากภาพที่ 4-5 ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ (cGA), ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าเฉลี่ย (mcGA) และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ (fb-cGA) สามารถหาคำตอบที่ดีที่สุดได้ โดยกราฟทางด้านซ้ายแสดงให้เห็นว่าค่าความน่าจะเป็นในแต่ละมิติของเวกเตอร์ความน่าจะเป็นสามารถที่จะวิ่งเข้าหาโทนสีขาวได้ทั้งหมด และกราฟทางด้านขวาก็แสดงถึงค่าความเหมาะสมที่สามารถวิ่งเข้าหาค่าที่สูงที่สุดได้อีกด้วย แต่ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับโดยใช้ตัวเก่งแบบถาวรและขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ โดยใช้ตัวเก่งแบบชั่วคราวไม่สามารถหาคำตอบที่ดีที่สุดได้ โดยกราฟทางด้านซ้ายแสดงให้เห็นถึงบางมิติของเวกเตอร์ความน่าจะเป็นที่ค่าความน่าจะเป็นกลายเป็นโทนสีดำ และกราฟทางด้านซ้ายแสดงถึงค่าความเหมาะสมที่ไม่สามารถวิ่งเข้าหาค่าที่ดีที่สุดได้

และจากภาพที่ 4-5 แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ (fb-cGA) สามารถที่จะหาคำตอบที่ดีที่สุดได้เร็วกว่าขั้นตอนวิธีอื่น ๆ ซึ่งตรงกันกับผลการทดลอง

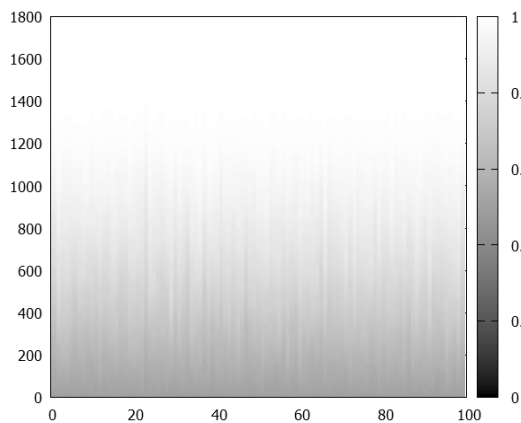
เปรียบเทียบประสิทธิภาพการทำงานของแต่ละขั้นตอนวิธีที่ได้แสดงก่อนหน้านี้ และแสดงให้เห็นถึงประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่ที่ซึ่งนำเสนอถึงการใช้เวลาในการหาคำตอบที่น้อย



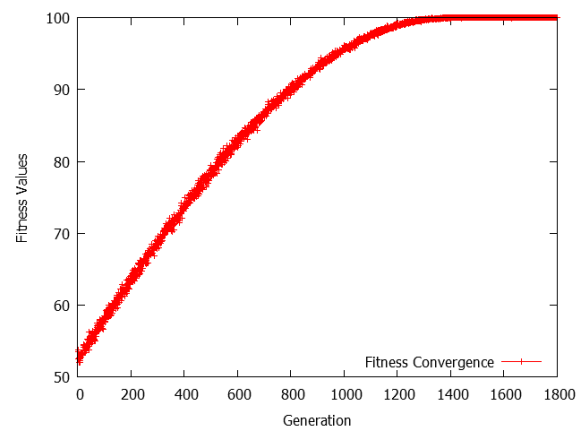
(ก) ค่าความน่าจะเป็น (cGA)



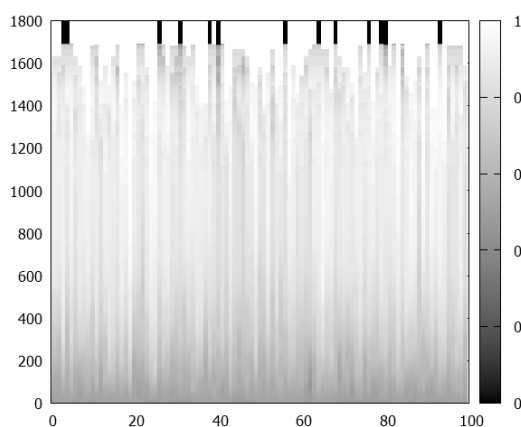
(ข) ค่าความเหมาะสม (cGA)



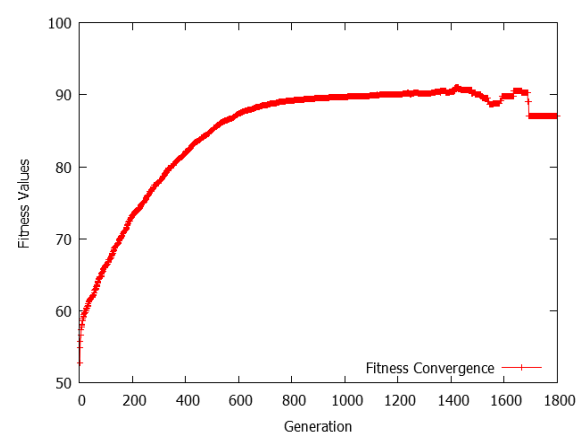
(ค) ค่าความน่าจะเป็น (mcGA)



(ง) ค่าความเหมาะสม (mcGA)

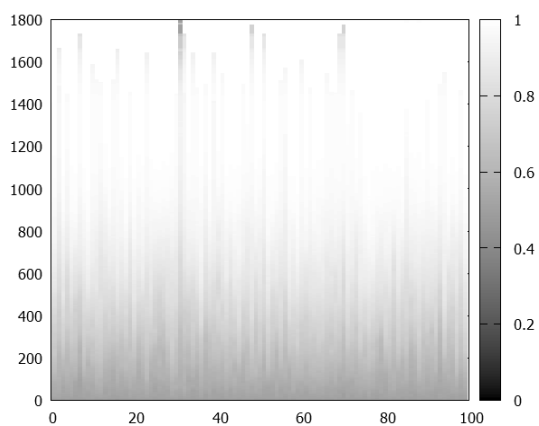


(จ) ค่าความน่าจะเป็น (pe-cGA)

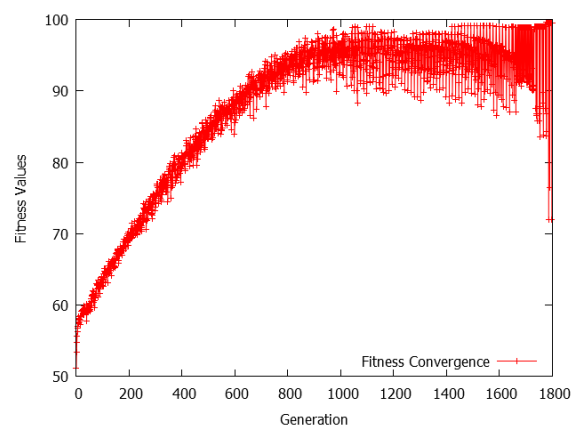


(ฉ) ค่าความเหมาะสม (pe-cGA)

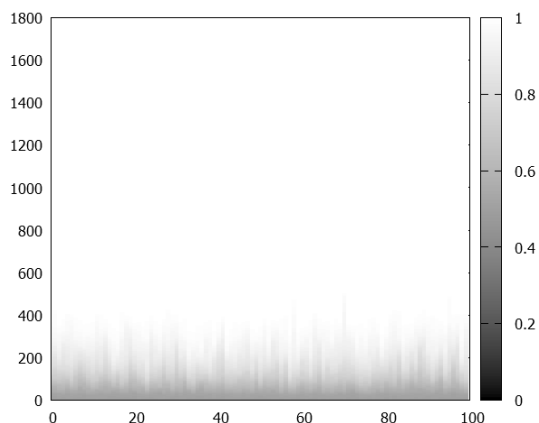
ภาพที่ 4-5 ค่าความน่าจะเป็น และค่าความเหมาะสม



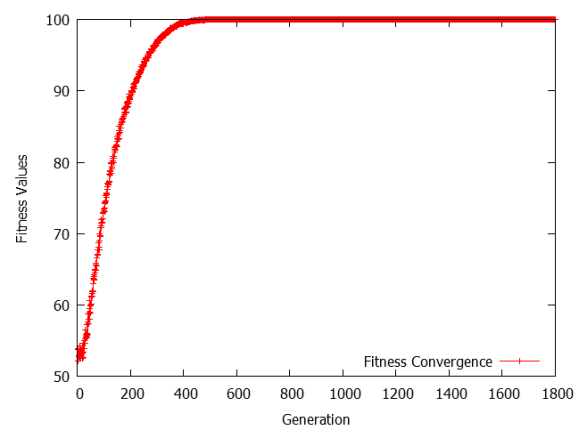
(ช) ค่าความน่าจะเป็น (ne-cGA)



(ซ) ค่าความเหมาะสม (ne-cGA)



(ฅ) ค่าความน่าจะเป็น (fb-cGA)



(ญ) ค่าความเหมาะสม (fb-cGA)

ภาพที่ 4-5 (ต่อ) ค่าความน่าจะเป็น และค่าความเหมาะสม

บทที่ 5

อภิปรายและสรุปผล

จากการศึกษาการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ผู้วิจัยได้พบกับปัญหาความไม่แน่นอนในการปรับปรุงค่าความน่าจะเป็น และกว่าที่ขั้นตอนวิธีจะมีทิศทางในการปรับปรุงค่าความน่าจะเป็นที่ชัดเจนจำนวนของวิวัฒนาการก็ได้เกิดขึ้นไปแล้วระยะหนึ่ง และจากการศึกษาค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้าย ผู้วิจัยก็พบว่าในการปรับปรุงค่าความน่าจะเป็นแต่ละรอบการทำงานมีค่าความถี่ และความต่อเนื่องเกิดขึ้น ซึ่งในมิติที่มีทิศทางในการปรับปรุงค่าความน่าจะเป็นที่ชัดเจนจะมีความถี่ในการปรับปรุงค่าความน่าจะเป็นไปในทิศทางนั้น ๆ อย่างต่อเนื่อง แต่ในมิติที่มีความไม่แน่นอนเกิดขึ้นมากนั้นจะไม่มีค่าความถี่ในการปรับปรุงค่าความน่าจะเป็นที่ต่อเนื่องไปยังทิศทางใดอย่างชัดเจน ดังนั้นผู้วิจัยจึงได้นำค่าความถี่ และค่าความต่อเนื่องนี้มาใช้ในการพัฒนาการปรับปรุงค่าความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ และให้ชื่อว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

ผลการวิจัย

งานวิจัยนี้เป็นงานวิจัยที่มีวัตถุประสงค์เพื่อปรับปรุงวิธีการปรับค่าแวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มขึ้น ซึ่งจากการศึกษาการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับผู้วิจัยได้พบปัจจัยที่สามารถจะนำมาใช้ปรับปรุงวิธีการปรับค่าแวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มขึ้นได้ออกเป็น 2 ปัจจัย

ปัจจัยที่หนึ่ง เนื่องจากผู้วิจัยได้พบกับปัญหาความไม่แน่นอนในการปรับปรุงค่าความน่าจะเป็น และกว่าที่ขั้นตอนวิธีจะมีทิศทางในการปรับปรุงค่าความน่าจะเป็นที่ชัดเจนจำนวนของวิวัฒนาการก็ได้เกิดขึ้นไปแล้วระยะหนึ่ง ผู้วิจัยจึงได้นำขนาดของประชากร ($psize$) มาหารด้วยจำนวน n แล้วได้เป็นอัตราส่วนของขนาดประชากร ($psize/n$) มาใช้ในการตัดสินใจปรับปรุงค่าความน่าจะเป็นด้วยค่าความถี่ที่ต่อเมื่อจำนวนรุ่นของวิวัฒนาการได้เกิดขึ้นมาถึงอัตราส่วนนี้แล้ว ซึ่งจากการทดลองวัดประสิทธิภาพโดยการแทนค่า n ด้วยค่า 2 3 4

และ 5 นั้น ผลการวัดประสิทธิภาพแสดงให้เห็นว่า $p_{size}/4$ ให้ค่าประสิทธิภาพที่ดีที่สุด ผู้วิจัยจึงได้ใช้อัตราส่วนนี้ในขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่

ปัจจัยที่สอง จากการศึกษาค่าความน่าจะเป็นที่เกิดขึ้นตั้งแต่การทำงานรอบแรกจนถึงการทำงานรอบสุดท้าย ผู้วิจัยก็พบว่าในการปรับปรุงค่าความน่าจะเป็นแต่ละรอบการทำงานมีค่าความถี่ และความต่อเนื่องของการปรับปรุงค่าความน่าจะเป็นในแต่ละรอบการทำงานเกิดขึ้น ซึ่งในมิติที่มีทิศทางในการปรับปรุงค่าความน่าจะเป็นเข้าหาค่า 0.0 หรือ 1.0 อย่างชัดเจนนั้นจะมีความถี่ และความต่อเนื่องในการปรับปรุงค่าความน่าจะเป็นไปในทิศทางนั้นอย่างมาก และสอดคล้องกัน ผู้วิจัยจึงได้นำค่าความถี่มาใช้ในการตัดสินใจปรับปรุงค่าความน่าจะเป็น และใช้ค่าความต่อเนื่องเป็นขนาดของการปรับปรุงค่าความน่าจะเป็น

จากผลการวิจัยปรากฏว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีประสิทธิภาพการหาคำตอบที่ดีที่สุดได้ ซึ่งสามารถเทียบได้กับขั้นตอนวิธีอื่น ๆ ที่มีผู้นำเสนอมาก่อนหน้า และยังใช้จำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยมาก ซึ่งเป็นผลการวิจัยที่แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีประสิทธิภาพการหาคำตอบที่รวดเร็ว และจากการทดลองเปรียบเทียบประสิทธิภาพกับขั้นตอนวิธีอื่น ๆ ที่มีผู้นำเสนอมาก่อนหน้ากับปัญหาที่ดัด ผลการทดลองก็แสดงให้เห็นว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยค่าความถี่มีจำนวนครั้งในการประเมินค่าความเหมาะสมที่น้อยกว่าขั้นตอนวิธีอื่นหลายเท่า

ข้อเสนอแนะ

ในการปรับปรุงวิธีการปรับค่าเวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มขึ้นนั้นอาจจะยังมีปัจจัยอื่นที่สามารถที่จะพัฒนาขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพเพิ่มมากขึ้นได้ ซึ่งในงานวิจัยนี้เป็นเพียงปัจจัยหนึ่งที่ผู้วิจัยนำมาใช้ในการปรับปรุงวิธีการปรับค่าเวกเตอร์ความน่าจะเป็น แต่จากการศึกษาวิจัยในช่วงแรกก็ยังคงมีความไม่แน่นอนเกิดขึ้นอยู่ ซึ่งอาจจะเป็นประโยชน์ในการนำไปพัฒนาต่อ เพื่อหาปัจจัยที่สามารถจะกำจัดปัญหาความไม่แน่นอนในการปรับปรุงค่าความน่าจะเป็นในช่วงแรกของวิวัฒนาการได้

บรรณานุกรม

- พินิจ อ่อนน้อย และ สุพจน์ เสงพระพรหม. (2557) การเพิ่มประสิทธิภาพขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับด้วยการปรับค่าเวกเตอร์ความน่าจะเป็นโดยค่านำหนักตามความเหมาะสม, การประชุมวิชาการระดับชาติ วลัยลักษณ์วิจัย ครั้งที่ 6, 3 - 4 กรกฎาคม 2557, มหาวิทยาลัยวลัยลักษณ์, นครศรีธรรมราช.
- อรวรรณ วิชาภูภาพร และคนอื่นๆ. (2548). *ขั้นตอนวิธีเชิงพันธุกรรมคอมพิวเตอร์แบบบีบอัด*. Bangkok (Thailand): ม.ป.พ.
- C. W. Ahn and R. S. Ramakrishna. (2003). *Elitism-based compact genetic algorithms*, in IEEE Transactions on Evolutionary Computation.
- D. E. Goldberg. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley.
- G. R. Harik. F. G. Lobo and D. E. Goldberg. (1999). *The compact genetic algorithm*. in IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4, 287-297.
- M.H. Afshar. (2009). *Application of a compact genetic algorithm to pipe network optimization problems*, in Transactions A: Civil Engineering, Vol. 16, No. 3, pp. 264-271.
- Sunisa Rimcharoen, Daricha Sutivong and Prabhas Chongstitvatana. (2006). *Updating Strategy in Compact Genetic Algorithm Using Moving Average Approach*, in IEEE Conference on Cybernetics and Intelligent Systems.

ภาคผนวก

ภาคผนวก ก

A Frequency-Based Updating Strategy in Compact Genetic Algorithm

Srichol Phiromlap, Sunisa Rimcharoen
 Faculty of Informatics, Burapha University, Chonburi 20131
 Email: vitscp@src.ku.ac.th, rsunisa@buu.ac.th

Abstract—The Compact Genetic Algorithm (cGA) is one of evolutionary algorithms. There are proofs in the literature that the cGA mimics the behavior of Simple Genetic Algorithm (sGA). The cGA has a benefit in requiring almost minimal memory to store candidate solutions. It represents a population as a probability distribution instead of storing whole candidate solutions. Although the cGA has many advantages, it has a limitation on solving some problems such as deceptive problem or so called trap function. Therefore, this paper proposes an adaptation of updating strategy in the compact genetic algorithm to help the algorithm to achieve a higher solution quality with fewer evaluations. We named the proposed technique as the frequency based compact genetic algorithm (fb-cGA). The fb-cGA employs information from the past. We count frequencies and continuity of updating probabilities for both up and down. The frequencies and continuity are used to guide an updating step size. The experiment results show that our proposed method requires fewer evaluations and achieves a higher solution quality than the cGA. It can save the number of fitness evaluations up to ninefold when compared with the cGA using tournament size of 2 on 3x10 trap problem.

Keywords: genetic algorithm, compact genetic algorithm, updating strategy, frequency based, trap function

I. INTRODUCTION

Genetic Algorithm (GA) [1, 2] is one of search algorithms inspired by natural evolution [3]. The search process of the GA begins with generating a pool of candidate solutions called a population. Each of candidate solutions is evaluated and a fitness value is assigned. The GA produces offspring by selecting good candidates and applies genetic operators such as crossover and mutation. The processes are repeated until the stopping criterion is fulfilled. Although the GA has been widely used in many applications, it has some limitation on requiring a large number of memories to store candidate solutions, which is difficult to apply to some fields such as an embedded hardware. Many years ago, Harik et al. [4] proposed the compact Genetic Algorithm (cGA). It represents the population as a probability distribution which is not require to store a whole set of candidate solutions. The cGA is proved that it achieves comparable quality with approximately the same number of fitness evaluations as the GA using a small amount of memory [4]. There are many research works that employ the cGA to solve a problem. For example, M.H. Afshar

[5] applied the cGA to pipe network optimization problems. Amr Badr et al. [6] used the elitism-based compact genetic algorithm to solved protein folding problem. Kathleen M. Timmerman [7] used the cGA to implement in an insect-scale flapping-wing micro air vehicle. However, for difficult problems such as deceptive problems, the cGA does not provide acceptable solutions. There are many attempts to modify and improve an updating strategy in the cGA. Bui Van Ha et al. [8] presented an improved cGA using simultaneously dealing with more probability vectors for EM complex system design. They used more than one probability vector (PV) to enhance the exploration properties of the algorithm. Sunisa Rimcharoen et al. [9] presented an improved cGA using moving average approach. Chang Wook Ahn and Ramakrishna [10] presented two elitism-based compact genetic algorithms namely persistent elitist compact genetic algorithm (pe-cGA), and nonpersistent elitist compact genetic algorithm (ne-cGA). The pe-cGA keeps the current best solution until a better solution is found and ne-cGA keeps the best solution for the life time.

In this paper, we propose using a frequency-based updating technique to modify the updating strategy of the cGA. The concept behind this technique is that we should make a use of information from the past. The number of updates in up and down directions is counted. It is used to adjust an updating step size toward the promising direction. The details will describe in section III.

The paper is organized as follows: Section II introduces background knowledge of the compact genetic algorithm. Section III describes the proposed technique. Experiment results and analysis are provided in Section IV. A conclusion is drawn in Section V.

II. THE COMPACT GENETIC ALGORITHM

The compact genetic algorithm (cGA) proposed by Harik, Lobo and Goldberg[4], employs a probabilistic model instead of using population in searching for solutions. This makes the compact genetic algorithm take advantage in using a small amount of memory and eliminating genetic operators such as crossover and mutation. This technique helps the algorithm to process faster.

The cGA represents a solution as a probability vector instead of a bit string. The number of dimensions of the vector equal to a chromosome length. Each dimension of the vector stores a probability that a bit is 1. It samples two candidate solutions from the probability vector. A fitness value is assigned to them. The winner and the loser are identified according to this value. The probability vector is then updated towards the winner. The pseudocode of cGA is shown in Fig. 1. The parameters are a population size($psize$), which is used to define an updating step size ($psize$ is analogous to population size in the simple genetic algorithm), and a chromosome length(l).

```

The pseudocode of cGA

Step 1: Initialize probability vector. Each dimension is
        set to 0.5

Step 2 : Create two candidate solutions by sampling from
        probability vector
        individual1 := generate(p);
        individual2 := generate(p);

Step 3 : Calculate fitness values of the two candidates

Step 4 : Compare the fitness values. The best one is a
        winner and the another is a loser

Step 5 : Update the probability vector according to the
        winner.

        for i:=1 to l
        begin
            if winner[i] ≠ loser[i] then
                if winner[i] = 1 then
                    p[i] := p[i] + 1/psize;
                else
                    p[i] := p[i] - 1/psize;
            end
        end

Step 6 : Check the vector whether it converges or not.
        If the probability vector converges (all
        dimensions are zero or one), the algorithm
        terminates. Otherwise, repeat step 2 to 6.

```

Figure 1. Pseudocode of cGA

First step, the cGA initializes the probability vector as 0.5 in every dimension. Two individuals are then generated from the vector denoted as *individual1* and *individual2* in the pseudocode . Next let *individual1* and *individual2* compete using their fitness values. The probability vector is updated towards the winner. The probability increases or decreases if the winner is not equal to the loser. It increases when the winner is one, and decreases when the winner is zero. The loop is repeated until the vector converges, that means all dimension of the probability vector contains zero or one.

Harik, Lobo and Goldberg also modified the cGA by adding more candidates, called a tournament cGA. The modified version samples candidate solutions of size s and uses a tournament selection to select the winner. By rearranging candidate solutions, denoted as an array $S[]$ in the pseudocode, $S[1]$ is the winner. The probability vector is updated by competing $S[1]$ and $S[j]$ in the same manner of step 5 in the original version.

Pseudocode of a tournament cGA

```

Step 1: Define tournament size of s

Step 2: Create s individuals from the probability vector
        and add them to S[i]

Step 3: Evaluate each individual

Step 4: Rearrange S so that S[i] is the best individual

Step 5: Compare S[i] with S[1] and update the
        probability vector
        for i := 2 to s do
        begin
            winner, loser := compete(S[1], S[i]);
            update the probability vector using step 5
            of the original cGA
        end

```

Figure 2. Pseudocode of a tournament cGA.

III. MODIFYING UPDATING METHOD

This paper proposes applying a frequency-based updating technique to update the probability vector in the compact genetic algorithm.

In each step, the cGA updates each dimension of the probability vector towards one or zero. Some dimensions rise and some are dropped. Some of them are also fluctuating. These show uncertainty in updating the probability vector. It is obvious that the probability vector fluctuates in the beginning period and converges to some direction in the end.

The information of these uncertainties that occur in updating process can use as a guideline to show a trend of converging to solution. Therefore, this paper proposes using a frequency and a continuation of updates to guide the algorithm to the promising direction by adjusting a suitable step size of the probability vector. In each dimension, we count the number of updates both of updating towards one and zero. We also count the number of continuity updates which perform in the same direction (called a continuation). The continuation is reset to zero if the vector is updated in an opposite direction. This proposed method shown in Fig. 3 which is a modification of step 5 of the cGA.

```

4) calculating the updating rate of fb-cGA

for  $i := 1$  to  $l$ 
begin
  if  $winner[i] \neq loser[i]$  then
    if  $winner[i] = 1$  then
       $Ufreq[i] = Ufreq[i] + 1;$ 
       $Ucon[i] = Ucon[i] + 1;$ 
       $Dcon[i] = 0;$ 
      if  $(Ufreq[i] > Dfreq[i] \text{ AND } Gen > (psize/3))$  then
         $p[i] := p[i] + ((1/psize) + (p[i] * (Ucon[i]/100)));$ 
      else
         $p[i] := p[i] + (1/psize);$ 
    else
       $Dfreq[i] = Dfreq[i] + 1;$ 
       $Dcon[i] = Dcon[i] + 1;$ 
       $Ucon[i] = 0;$ 
      if  $(Dfreq[i] > Ufreq[i] \text{ AND } Gen > (psize/3))$  then
         $p[i] := p[i] + ((1/psize) + (p[i] * (Dcon[i] / 100)));$ 
      else
         $p[i] := p[i] + (1/psize);$ 
end

```

The parameters :

- $Ufreq$: the number of increment updates.
- $Dfreq$: the number of decrement updates.
- $Ucon$: the number of continuous increment updates.
- $Dcon$: the number of continuous decrement updates.
- Gen : the number of generations.

Figure 3. Pseudocode of a modification of cGA.

Fig. 3 presents a pseudocode of the frequency-based updating strategy in the compact genetic algorithm (fb-cGA). The probability vector is updated when the winner and the loser are not equal. In case that the winner is one, the algorithm checks whether $Ufreq$ is greater than $Dfreq$ or not. If $Ufreq$ is greater than $Dfreq$, it means that in this dimension the probability vector should be updated according to the majority, the algorithm offers larger step size. In this paper, we also use the confident value, the number of updates that count from a continuity of updating in the same direction, to make a larger updating step size. The algorithm performs in the same manner. When the winner is zero, the algorithm updates probability vector by decreasing probability towards zero. Note that the proposed strategy is performed when the generation is greater than $1/3$ of population size($psize$). It uses the original method in the early generation. This is because we should give enough time and let the algorithm runs and explores solutions before using past information to make a decision.

IV. EXPERIMENT RESULT AND ANALYSIS

In the first experiments, we test the proposed method with one-max problem, which is an easy toy problem to test genetic algorithm. The objective of this problem is to find a bit string that contains all one. The fitness value is given according to the number of 1-bits in an individual. Fig. 4 shows the results of

100 bits one-max when varying a population size from 2, 4, 6, 8 to 100 with tournament size 2. The graph is shown with average results from 50 runs. The y-axis is average values of solutions quality (the number of correct bits). The x-axis is population sizes. We compare the results of fb-cGA with sGA, cGA and mcGA. The result of the fb-cGA is comparable with other techniques.

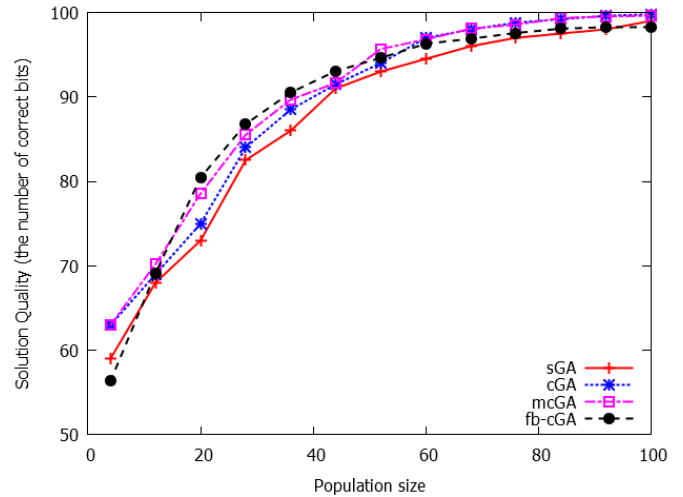


Figure 4. Solution quality on the one-max problem.

The trap problem, the difficult function to test the genetic algorithm, is designed to fool gradient-based optimizers to favor zeroes, but the optimal solution is composed of all ones.

The graph in Fig. 5 shows fitness values of 3-bits trap function. The maximum fitness value obtained when all bits are one. But, when the candidate solution contains two bits of 1, the fitness value is zero.

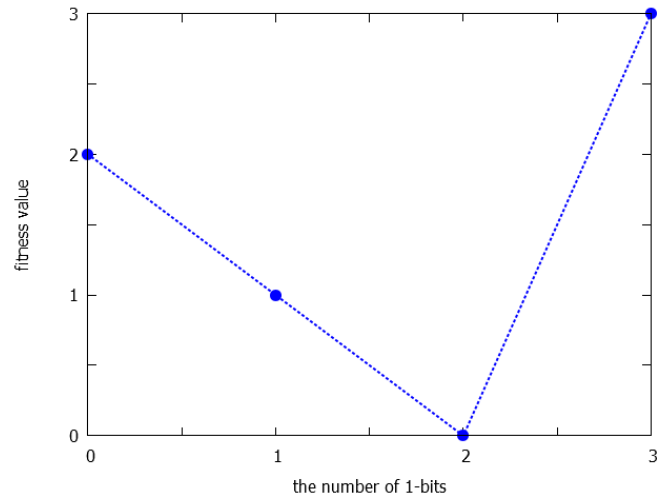


Figure 5. 3-bits Trap Function

We can create $k \times m$ trap problem by combining in groups of k -bits trap, and the fitness value is calculated by summing up the score of all groups. Here, we test the algorithm with 3×10

trap problem. For example, a candidate solution “111 001 110 000 111” is assigned a fitness value of 9.

Fig. 6 and Fig. 7 show the results on trap problem using tournament size of 2 and population sizes of 8, 500, 1000, 1500, 2000, 2500 and 3000. Fig. 6 shows solution quality as the number of correct building blocks. The result from the fb-cGA has higher quality of solution than sGA, cGA and mcGA. However, the pe-cGA and ne-cGA outperform other techniques in terms of solution quality. Fig. 7 shows the number of function evaluations used to find a solution. The experiment results show that the fb-cGA requires smallest number of fitness evaluations. It uses fewer fitness evaluations than the sGA about fourteen times, the original cGA about nine times, the mcGA about twelve times, the pe-cGA about three times and the ne-cGA about two times. These show the efficiency of the proposed method in terms of saving fitness evaluations.

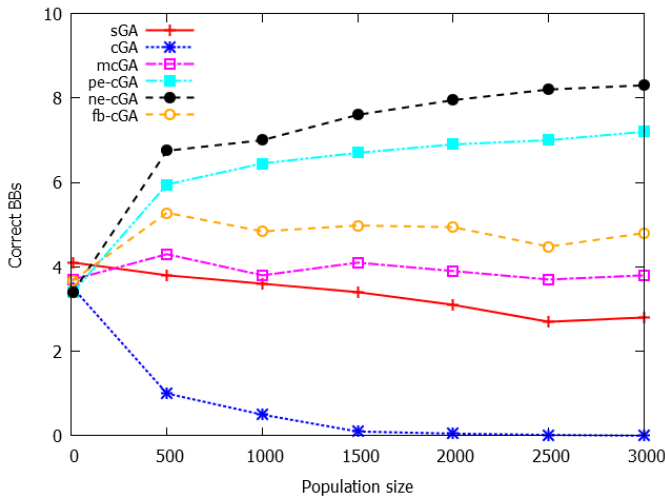


Figure 6. Solution quality on the trap problem with tournament size of 2.

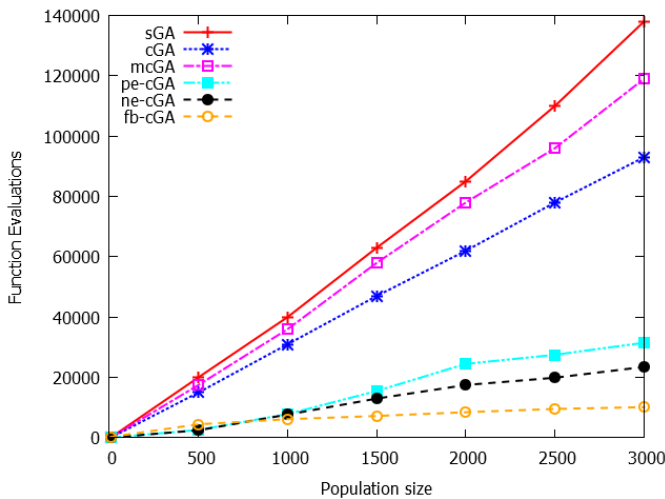


Figure 7. The number of function evaluations on the trap problem with tournament size of 2.

Fig. 8 and Fig. 9 show the results on trap problem using tournament size of 4 and population sizes of 8, 500, 1000, 1500, 2000, 2500 and 3000. Figure VII shows solution quality as the number of correct building blocks. The result from the fb-cGA has higher quality of solution than pe-cGA and cGA. However, the sGA, mcGA and ne-cGA outperform other techniques in terms of solution quality. Figure VIII shows the number of function evaluations used to find a solution. The experiment results show that the fb-cGA requires smallest number of fitness evaluations.

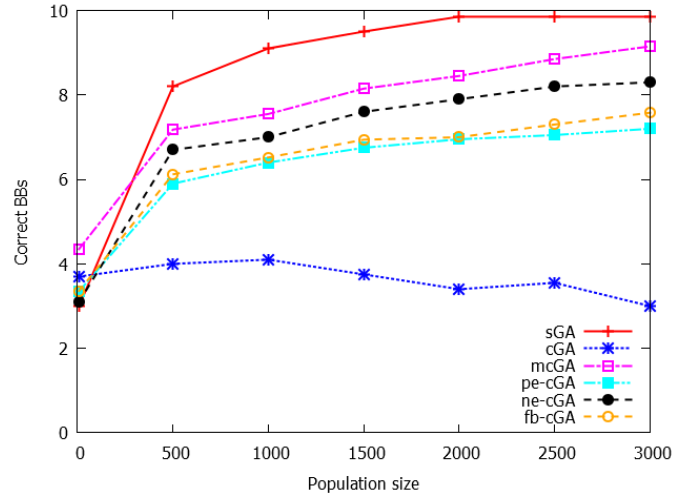


Figure 8. Solution quality on the trap problem with tournament size of 4.

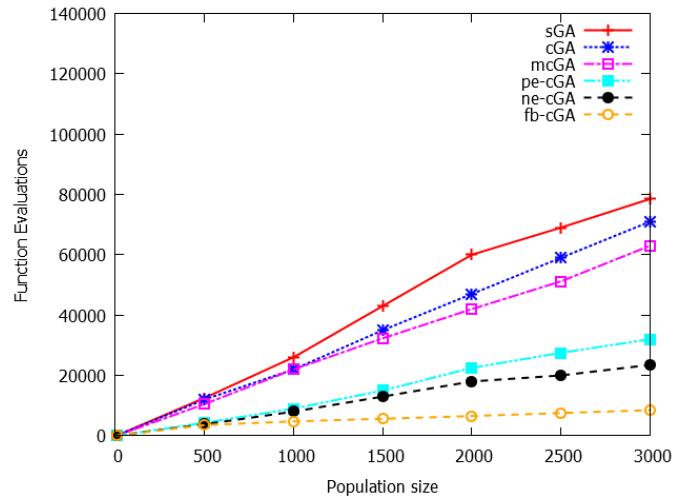


Figure 9. The number of function evaluations on the trap problem with tournament size of 4.

Fig. 10 and Fig. 11 show the results on trap problem using tournament size of 8 and population sizes of 8, 500, 1000, 1500, 2000, 2500 and 3000. Figure IX shows solution quality as the number of correct building blocks. The result from the fb-cGA has higher quality of solution than cGA, pe-cGA and ne-cGA. However, the sGA and mcGA outperform other techniques in terms of solution quality. Figure X shows the number of function evaluations used to find a solution. The experiment results show that the fb-cGA requires smallest number of fitness evaluations.

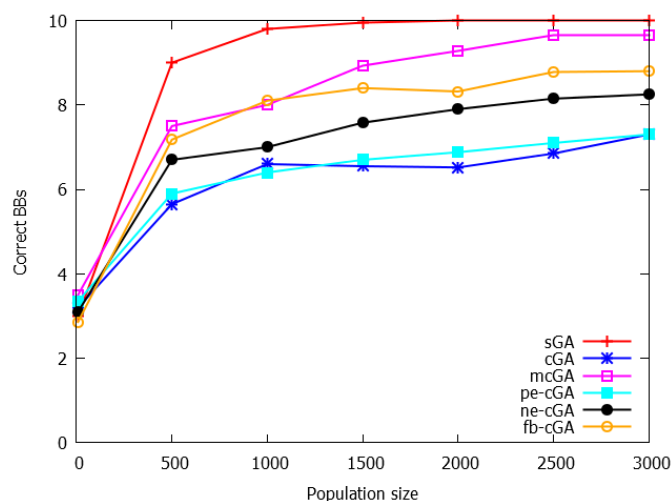


Figure 10. Solution quality on the trap problem with tournament size of 8.

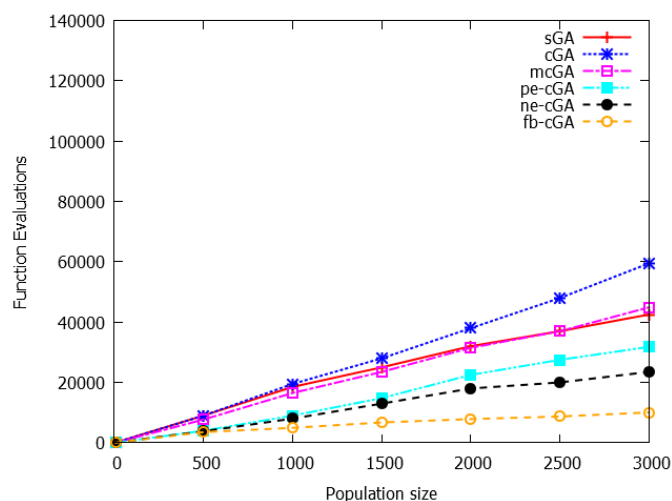


Figure 11. The number of function evaluations on the trap problem with tournament size of 8.

V. CONCLUSIONS

This paper proposes using a frequency-based updating approach to modify the updating strategy of the cGA. The proposed method takes the advantage of the frequency and continuity of increasing and decreasing probability to modify an updating step size. The experiments on simple and deceptive problems show that the frequency-based updating technique can yield a comparable result with other techniques in the one-max problem and can improve the solution quality with a smaller number of function evaluations than the original cGA in the trap problem.

REFERENCES

- [1] D. E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley, 1989.
- [2] J. H. Holland, Adaptation in natural and artificial systems, University of Michigan Press, 1975.
- [3] C. Darwin, On the Origin of Species by means of Natural Selection, John Murray, 1859.
- [4] G. R. Harik, F. G. Lobo and D. E. Goldberg, The compact genetic algorithm, in IEEE Transactions on Evolutionary Computation, 1999, Vol. 3, No. 4, 287-297.
- [5] M.H. Afshar, Application of a compact genetic algorithm to pipe network optimization problems, in Transactions A: Civil Engineering, 2009, Vol. 16, No. 3, pp. 264-271.
- [6] Amr Badr, Ibtehal M. Aref, Basma M. Hussien and Yosr Eman, Solving protein folding problem using elitism-based compact genetic algorithm, in Journal of Computer Science 4 (7): 525-529, 2008, ISSN 1549-3636.
- [7] Kathleen M. Timmerman, A hardware compact genetic algorithm for hover improvement in an insect-scale flapping-wing micro air vehicle, A thesis for the degree of master of science in computer science, Wright State University, 2010.
- [8] Bui Van Ha, R.E. Zich, M. Mussetta, Paola Pirinoli and Chien Ngoc Dao, Improved compact genetic algorithm for EM complex system design, in IEEE Conference on Communications and Electronics (ICCE), 2012.
- [9] Sunisa Rimcharoen, Daricha Sutivong and Prabhas Chongstitvatana, Updating Strategy in Compact Genetic Algorithm Using Moving Average Approach, in IEEE Conference on Cybernetics and Intelligent Systems, 2006.
- [10] C. W. Ahn and R. S. Ramakrishna, Elitism-based compact genetic algorithms, in IEEE Transactions on Evolutionary Computation, 2003.

ภาคผนวก ข

ที่ ๑๗๙/๒๕๕๖



เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาโครงร่างวิจัย
เรื่อง วิธีการปรับปรุงเวกเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระตุ้นด้วยค่าความถี่

หัวหน้าโครงการวิจัย นายศรีชล ภิรมย์ลาภ

หน่วยงานที่สังกัด นิสิตระดับบัณฑิตศึกษา คณะวิทยาการสารสนเทศ

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาแล้วเห็นว่า
เค้าโครงวิทยานิพนธ์ดังกล่าวเป็นไปตามหลักการของจริยธรรมการวิจัยในมนุษย์ โดยที่ผู้วิจัยเคารพสิทธิและศักดิ์ศรี
ในความเป็นมนุษย์ ไม่มีการล่วงละเมิดสิทธิ สวัสดิภาพ และไม่ก่อให้เกิดภัยอันตรายแก่ตัวอย่างการวิจัยกลุ่มตัวอย่าง
และผู้เข้าร่วมในโครงการวิจัย

จึงเห็นสมควรให้ดำเนินการวิจัยในขอบข่ายของโครงร่างวิทยานิพนธ์ที่เสนอได้ ตั้งแต่วันที่ออกเอกสาร
รับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์ฉบับนี้จนถึงวันที่ ๓๑ เดือนตุลาคม พ.ศ. ๒๕๕๖

ออกให้ ณ วันที่ ๔ ตุลาคม พ.ศ. ๒๕๕๖

ลงนาม

(ผู้ช่วยศาสตราจารย์ ดร.สมถวิล จริตควร)

ประธานคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา