ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับและการประยุกต์
Compact Genetic Algorithm and Its Applications

สุนิสา ริมเจริญ*
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
Sunisa Rimcharoen*
Faculty of Informatics, Burapha University.

บทคัดย่อ

บทความนี้นำเสนอความรู้เบื้องต้นและการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ซึ่งเป็นขั้นตอนวิธีที่อาศัยหลักการวิวัฒนาการคำตอบคล้ายกับขั้นตอนวิธีเชิงพันธุกรรมที่มีการใช้งานกันอย่างแพร่หลายในปัจจุบัน การนำขั้นตอนวิธีเชิงพันธุกรรมแบบเดิมไปใช้ในฮาร์ดแวร์ขนาดเล็กเป็นเรื่องยาก เนื่องจากขั้นตอนวิธีเชิงพันธุกรรมต้องอาศัยประชากรจำนวนมากในการหาคำตอบ และต้องการฮาร์ดแวร์ที่มีความสามารถในการประมวลผลค่อนข้างสูง

วิธีเชิงพันธุกรรมแบบกระชับที่นำเสนอนี้มีลักษณะเด่นในการใช้หน่วยความจำที่เกือบจะน้อยที่สุดในการเก็บตัวอย่างคำตอบที่เป็นไปได้ เนื่องจากขั้นตอนวิธีนี้ใช้วิธีการแทนโครงสร้างของประชากรด้วยเวคเตอร์ความน่าจะเป็นที่แจกแจงความน่าจะเป็นของคำตอบ งานวิจัยที่ตีพิมพ์ก่อนหน้านี้ก็มีการพิสูจน์ว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับมีพฤติกรรมการทำงานที่สามารถเทียบเคียงได้กับพฤติกรรมการทำงานของขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายที่ใช้การไขว้เปลี่ยนแบบเอกรูป แต่ใช้หน่วยความจำน้อยกว่า ทั้งสองขั้นตอนวิธีให้คำตอบที่มีคุณภาพใกล้เคียงกันเมื่อใช้จำนวนครั้งที่เท่ากันในการประเมินค่าความเหมาะสม ทำให้ขั้นตอนวิธีนี้มีข้อได้เปรียบและเหมาะต่อการนำไปประยุกต์ใช้ในงานที่มีข้อจำกัดในเรื่องขนาดของหน่วยความจำและความสามารถในการประมวลผล

คำสำคัญ : ขั้นตอนวิธีเชิงพันธุกรรม ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ การคำนวณเชิงวิวัฒนาการ ขั้นตอนวิธีประมาณการแจกแจง ขั้นตอนวิธีเชิงพันธุกรรมแบบการสร้างแบบจำลองความน่าจะเป็น

Abstract

This paper presents an introduction and applications of the Compact Genetic Algorithm (cGA), one of evolutionary algorithms similar to genetic algorithms widely used to solve current real-world problems. Executing traditional genetic algorithms on a small-sized hardware is ineffective due to large number of population and high processing power requirement.

The cGA has a distinct characteristic that it requires almost minimal memory to store candidate solutions. It represents a population structure as a probability distribution over the set of solutions. There are proofs in the literature that the cGA mimics the behavior of Simple Genetic Algorithm (sGA) with uniform crossover using a small amount of memory, and achieves comparable quality with approximately the same number of fitness evaluations. Thus, these advantages contribute to a flexible implementation for the problems that have limitations on memory usage and computational resources.

Keywords : Genetic Algorithm, Compact Genetic Algorithm, Evolutionary Computation, Estimation of Distribution Algorithm, Probabilistic Model Building Genetic Algorithm

***E-mail:** rsunisa@buu.ac.th

บทนำ

ขั้นตอนวิธีเชิงพันธุกรรมเป็นขั้นตอนวิธีการหาคำตอบที่เลียนแบบกระบวนการวิวัฒนาการตามธรรมชาติ โดย Darwin (1859) ได้พูดถึงทฤษฎีวิวัฒนาการไว้ว่า สิ่งมีชีวิตจะมีการถ่ายทอดลักษณะทางพันธุกรรมจากรุ่นสู่รุ่น เพื่อที่จะดำรงเผ่าพันธุ์อยู่ต่อไป สิ่งมีชีวิตที่สามารถปรับตัวได้เหมาะสมกับสภาพแวดล้อมก็จะอยู่รอดและสืบลูกหลาน ส่วนสิ่งมีชีวิตใดที่อ่อนแอก็จะล้มตายลง วิธีการคัดเลือกตามธรรมชาติที่ให้สิ่งมีชีวิตที่เหมาะสมอยู่รอดและสืบเผ่าพันธุ์ต่อนี้ ทำให้เกิดกระบวนการวิวัฒนาการขึ้น สิ่งมีชีวิตต่างๆ ต้องพยายามปรับตัวเองให้เข้มแข็ง เพื่อที่จะผ่านการคัดสรรตามธรรมชาติ และเมื่อเวลาผ่านไป สิ่งมีชีวิตที่ดีและเหมาะสมก็จะเกิดขึ้น

หลักการวิวัฒนาการตามธรรมชาติข้างต้นทำให้เกิดแรงบันดาลใจในการพัฒนาวิธีการคำนวณเชิงวิวัฒนาการขึ้นมา ในช่วงทศวรรษที่ 1960 ถึง 1970 นักวิจัยหลายกลุ่มได้เริ่มพัฒนาเทคนิคการคำนวณโดยอาศัยหลักการนี้ สามวิธีการหลักๆ ที่ถูกนำเสนอในยุคนั้น คือ การโปรแกรมเชิงวิวัฒน์ (Evolutionary programming) กลยุทธ์เชิงวิวัฒน์ (Evolution strategy) และขั้นตอนวิธีเชิงพันธุกรรม (Genetic algorithm)

การโปรแกรมเชิงวิวัฒน์นำเสนอโดย Fogel *et al*. (1965, 1966) เพื่อใช้วิวัฒนาการเครื่องจักรสถานะแบบจำกัด (Finite state machine) ในเวลาใกล้เคียงกันนักวิจัยที่เยอรมันได้นำเสนอกลยุทธ์เชิงวิวัฒน์ (Rechenberg, 1965, 1971; Schwefel, 1974) เพื่อหาค่าเหมาะสุดเชิงตัวเลข (Numerical optimization) ทั้งสองวิธีนี้มีความใกล้เคียงกันมากและมีข้อเด่นในการประยุกต์ใช้กับข้อมูลที่เป็นจำนวนจริง แต่ในปัจจุบันขั้นตอนวิธีที่ถูกนำมาใช้อย่างแพร่หลายคือขั้นตอนวิธีเชิงพันธุกรรม ซึ่งนำเสนอโดย Holland (1973, 1975) และมาเป็นที่รู้จักในวงกว้างจากหนังสือของ Goldberg (1989)

ผลเฉลยหรือคำตอบที่เป็นไปได้ของปัญหาต่างๆ ในขั้นตอนวิธีเชิงพันธุกรรมจะถูกแทนด้วยรูปแบบโครโมโซม (Chromosome) แต่ละโครโมโซมถูกแทนด้วยสายอักขระของเลขฐานสอง ซึ่งในหนึ่งประชากร (Individual) อาจมีมากกว่า 1 โครโมโซมก็ได้ และหลายๆ ประชากรรวมกันจะกลายเป็นกลุ่มประชากร (Population) ที่ใช้ในการวิวัฒนาการไปสู่คำตอบที่ดีขึ้น

การทำงานของขั้นตอนวิธีเชิงพันธุกรรมจะเริ่มต้นจากการสุ่มประชากรขึ้นมาจำนวนหนึ่ง แล้วประเมินค่าความเหมาะสมของประชากรแต่ละตัว โดยการแปลงสายอักขระที่ถูกจัดเก็บไว้เพื่อหาค่าความเหมาะสม (Fitness value) โดยคำนวณจากฟังก์ชันหาค่าความเหมาะสม (Fitness function) ซึ่งจะแตกต่างกันไปขึ้นกับชนิดของปัญหา ประชากรที่ดีในกลุ่มประชากรที่ถูกประเมินค่าแล้วอาจจะถูกเลือกและผ่านการดำเนินการทางพันธุกรรม เช่น การไขว้เปลี่ยน การกลายพันธุ์ เพื่อสร้างกลุ่มประชากรรุ่นใหม่ แต่ละครั้งที่สร้างกลุ่มประชากรรุ่นใหม่ขึ้นมาจะเรียกว่า “รุ่น” (Generation) กระบวนการวิวัฒนาการจะเลือกประชากรที่ดีในแต่ละรุ่นที่จะอยู่รอดในรุ่นต่อไป และจะวนซ้ำอยู่อย่างนี้จนกว่าจะถึงเงื่อนไขการหยุดที่กำหนดไว้

แนวคิดของขั้นตอนวิธีเชิงพันธุกรรมได้มีการพัฒนาเรื่อยมาจนปัจจุบันนี้ได้มีการเสนอมุมมองใหม่ในการแทนคำตอบด้วยการประมาณการแจกแจงความน่าจะเป็น ซึ่งแทนที่จะแทนคำตอบด้วยกลุ่มประชากร ก็จะแทนคำตอบด้วยตัวแบบความน่าจะเป็นแทน ขั้นตอนวิธีประมาณการแจกแจงนี้จะไม่มีกลุ่มประชากร ไม่มีการดำเนินการทางพันธุกรรม เช่น การไขว้เปลี่ยน การกลายพันธุ์อีกต่อไป แต่จะใช้วิธีการสร้างและปรับปรุงตัวแบบของคำตอบจากตัวอย่างดีๆ ที่ได้สุ่มเลือกขึ้นมา

Mühlenbein & Paaß (1996); Pelikan *et al*. (1999) นำเสนอขั้นตอนวิธีประมาณการแจกแจง (Estimation of distribution algorithms: EDAs) หรือขั้นตอนวิธีเชิงพันธุกรรมแบบการสร้างแบบจำลองความน่าจะเป็น (Probabilistic model building genetic algorithms: PMBGAs) โดยในขั้นตอนวิธีนี้ตัวแปรในแต่ละมิติ (อาจมองเทียบได้กับแต่ละบิตในขั้นตอนวิธีเชิงพันธุกรรมแบบเดิม) จะมีค่าความน่าจะเป็นในการเป็นคำตอบ รวมทั้งความสัมพันธ์กับตัวแปรอื่น และจะใช้การปรับค่าความน่าจะเป็นเพื่อปรับปรุงการกระจายตัวของคำตอบให้ไปในทิศทางที่ผลเฉลยมีค่าความเหมาะสมสูงขึ้น

ขั้นตอนวิธีประมาณการแจกแจงสามารถแบ่งได้เป็น 3 กลุ่มหลักๆ ตามลักษณะการขึ้นต่อกันของตัวแปร คือ ขั้นตอนวิธีประมาณการแจกแจงแบบที่ตัวแปรไม่ขึ้นต่อกัน (Univariate estimation of distribution algorithm) ขั้นตอนวิธีประมาณการแจกแจงแบบที่ตัวแปรขึ้นต่อกันเป็นคู่ (Bivariate estimation of distribution algorithm) และขั้นตอนวิธีประมาณการแจกแจงแบบตัวแปรหลายตัวขึ้นต่อกัน (Multivariate estimation of distribution algorithm) ในบทความนี้ ผู้เขียนจะพูดถึงเฉพาะขั้นตอนวิธีประมาณการแจกแจงแบบง่ายที่สุดคือไม่มีความขึ้นต่อกันในตัวแปรแต่ละตัว

ตัวอย่างของขั้นตอนวิธีประมาณการแจกแจงแบบที่ตัวแปรไม่ขึ้นต่อกัน เช่น การเรียนรู้เพิ่มขึ้นแบบอาศัยประชากร

(Population based incremental learning: PBIL) ซึ่งถูกเสนอโดย Baluja (1994) ขั้นตอนวิธีแจกแจงตามขอบหนึ่งตัวแปร (Univariate marginal distribution algorithm: UMDA) ซึ่งถูกเสนอโดย โดย Mühlenbein & Paaß (1996) และขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ (Compact genetic algorithm: cGA) ซึ่งถูกเสนอโดย Harik *et al*. (1999) ทั้ง 3 ขั้นตอนวิธีนี้มีหลักการร่วมกันคือ ใช้เวคเตอร์ความน่าจะเป็นแทนตัวแบบของคำตอบ โดยมีวิธีการปรับค่าความน่าจะเป็นที่แตกต่างกันไป ในบทความนี้ ผู้เขียนได้เลือกที่จะกล่าวถึงขั้นตอนวิธีที่น่าสนใจและมีการนำไปประยุกต์ใช้อย่างแพร่หลาย นั่นคือขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ซึ่งจะกล่าวรายละเอียดในหัวข้อถัดไป

ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

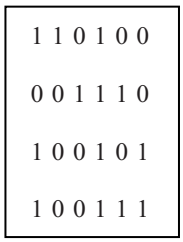
ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ เป็นหนึ่งในขั้นตอนวิธีเชิงวิวัฒนาการแบบใหม่ ที่มีแนวความคิดในการใช้ตัวแบบความน่าจะเป็น (Probabilistic model) แทนการใช้กลุ่มประชากรแบบเดิมในการค้นหาคำตอบ แนวความคิดนี้ทำให้ขั้นตอนวิธีเชิงพันธุกรรมใช้หน่วยความจำในการเก็บประชากรน้อยลง เนื่องจากไม่มีการใช้ประชากรอีกต่อไป อีกทั้งยังไม่ต้องอาศัยการดำเนินการเชิงพันธุกรรม เช่น การไขว้เปลี่ยน หรือ การกลายพันธุ์ ทำให้การประมวลผลทำได้รวดเร็วยิ่งขึ้น โดยที่ยังคงความสามารถเทียบเท่ากับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายที่ใช้อยู่เดิม ซึ่งข้อพิสูจน์นี้ปรากฏในงานวิจัยของ Harik *et al*. (1999)

การแทนคำตอบของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะอยู่ในรูปแบบของเวคเตอร์ความน่าจะเป็น (Probability vector) ซึ่งจะใช้เป็นตัวแบบในการหาการกระจายตัวของคำตอบ โดยแต่ละมิติ (Dimension) ของเวคเตอร์เป็นค่าความน่าจะเป็นที่แต่ละบิตจะเป็น 1 ตัวอย่างเช่น สมมุติว่าประชากรของขั้นตอนวิธีเชิงพันธุกรรมแบบเดิมมีโครโมโซมยาว 6 บิตดังภาพที่ 1(ก) ตัวอย่างของเวคเตอร์ความน่าจะเป็นของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับอาจเป็นดังภาพ 1(ข) ซึ่งจะมีจำนวนมิติคือ 6 ตามความยาวของโครโมโซมในขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย จากภาพที่ 1 จะเห็นได้ว่าการแทนคำตอบโดยใช้เวคเตอร์ความน่าจะเป็นนี้ สามารถลดหน่วยความจำที่ใช้จัดเก็บประชากรลงไปได้อย่างมาก เช่นจากตัวอย่างนี้ แทนที่จะใช้หน่วยความจำในการเก็บประชากร 4 ตัว ก็จะเหลือเพียงเวคเตอร์จำนวนจริงเพียงเวคเตอร์เดียว โดยในเวคเตอร์ความน่าจะเป็นนี้ แต่ละมิติจะเป็นค่าความน่าจะเป็นของการเกิดบิตที่เป็น 1 เช่น ค่า 0.75 แทนความน่าจะเป็นที่โครโมโซมบิตแรกจะเป็น 1 จากประชากรทั้งหมด เป็นต้น

การปรับปรุงคำตอบของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ สามารถทำได้โดยการปรับค่าความน่าจะเป็นในแต่ละมิติของเวคเตอร์ความน่าจะเป็นตามคำตอบที่ดีกว่า โดยในการทำงานของอัลกอริทึมนี้จะมีการสุ่มสร้างตัวอย่าง 2 ตัว ขึ้นมาจากเวคเตอร์ความน่าจะเป็น โดยโครโมโซมของตัวอย่างที่สุ่มขึ้นมานี้ในแต่ละบิตจะเป็น 0 หรือ 1 ขึ้นกับค่าความน่าจะเป็นในแต่ละมิติของเวคเตอร์ความน่าจะเป็น เช่น ถ้าเวคเตอร์ความน่าจะเป็นมีค่า 0.5 ในทุกมิติ ตัวอย่างประชากรที่สุ่มสร้างขึ้นมา จะมีโอกาสเป็นบิต 1 หรือ 0 เท่าๆ กัน แต่ถ้าเวคเตอร์ความน่าจะเป็นมีค่า 1.0 ทุกมิติ ตัวอย่างที่สุ่มออกมาได้จะเป็นโครโมโซมที่เป็นบิต 1 ทั้งหมด

เมื่อสุ่มสร้างตัวอย่างขึ้นมาแล้วก็จะพิจารณาว่าคำตอบตัวใดมีค่าความเหมาะสมมากกว่ากัน เวคเตอร์ความน่าจะเป็นในแต่ละมิติจะถูกปรับตามบิตของคำตอบตัวที่ดีกว่า ถ้าบิตนั้นมีค่าเป็น 1 ความน่าจะเป็นในมิตินั้นก็จะปรับเข้าใกล้ 1 แต่ถ้าเป็น 0 ค่าความน่าจะเป็นในมิตินั้นก็จะลดค่าลง โอกาสที่บิตนั้นจะถูกสร้างมาเป็น 0 ในรอบถัดไปก็จะเพิ่มตามด้วย ดังนั้นเมื่อผ่านกระบวนการวิวัฒนาการไประยะหนึ่ง เวคเตอร์ความน่าจะเป็นก็จะเป็นรูปแบบการกระจายตัวของคำตอบที่ดี โดยการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับมีขั้นตอนโดยสรุปดังนี้

1. กำหนดค่าเริ่มต้นของตัวแปรในเวคเตอร์ความน่าจะเป็นในทุกๆ มิติให้มีค่าเป็น 0.5



1(ก)

0.75	0.25	0.25	1.0	0.5	0.5

1(ข)

ภาพที่ 1 การแทนประชากรของขั้นตอนวิธีเชิงพันธุกรรมเทียบกับเวคเตอร์ความน่าจะเป็น

2. สุ่มสร้างตัวอย่างคำตอบจากเวคเตอร์ความน่าจะเป็นขึ้นมา 2 ตัว

3. คำนวณค่าความเหมาะสมของตัวอย่างที่สุ่มมาได้ แล้วพิจารณาว่าตัวใดเป็นผู้ชนะและผู้แพ้

4. ปรับค่าเวคเตอร์ความน่าจะเป็นตามผู้ชนะ โดยพิจารณาเฉพาะบิตที่ผู้ชนะและผู้แพ้ไม่เหมือนกัน

- ถ้าบิตตำแหน่งที่ *i* ของผู้ชนะ เป็น 1 จะปรับค่าเวคเตอร์ความน่าจะเป็นที่ตำแหน่ง *i* โดยนำค่าความน่าจะเป็นเดิมบวกกับ 1/*n* เมื่อ *n* คือจำนวนประชากรที่ถูกกำหนดใช้ในขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย
- ในทางกลับกัน ถ้าบิตตำแหน่งที่ *i* ของผู้ชนะ เป็น 0 จะปรับค่าเวคเตอร์ความน่าจะเป็นที่ตำแหน่ง *i* โดยนำค่าความน่าจะเป็นเดิมลบด้วย 1/*n*

5. ตรวจสอบว่าเวคเตอร์ความน่าจะเป็นลู่เข้าสู่คำตอบแล้วหรือไม่ โดยพิจารณาจากค่าความน่าจะเป็นของแต่ละมิติว่าเป็น 0.0 หรือ 1.0 ครบแล้วหรือไม่ ถ้าค่าความน่าจะเป็นลู่เข้าสู่ 0.0 หรือ 1.0 หมดแล้ว ให้จบการทำงาน แต่ถ้ายังลู่เข้าไม่ครบ ให้สุ่มตัวอย่างจากเวคเตอร์ความน่าจะเป็นขึ้นมาใหม่แล้วทำซ้ำขั้นตอนที่ 2 ถึง 5 ต่อไป

ลำดับต่อไปเป็นการแสดงตัวอย่างการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับเมื่อนำมาประยุกต์ใช้กับปัญหาจำนวนบิตหนึ่งมากสุด (One-Max problem) ปัญหานี้เป็นปัญหาสมมุติ (Toy problem) และเป็นปัญหาที่แก้ง่ายเมื่อใช้ขั้นตอนวิธีเชิงพันธุกรรม และมักจะถูกใช้เป็นปัญหาทดสอบพฤติกรรมของอัลกอริทึมเพื่อเปรียบเทียบความสามารถในการแก้ปัญหาแบบง่าย โดยค่าความเหมาะสมของคำตอบจะเท่ากับจำนวนบิตที่เป็น 1 ถ้ามีบิตที่เป็น 1 ยิ่งมากค่าความเหมาะสมจะมากขึ้น และค่าสูงสุดที่เป็นไปได้จะเกิดในกรณีที่ทุกบิตเป็น 1 ทั้งหมด ในกรณีนี้ค่าความเหมาะสมจะเท่ากับความยาวของโครโมโซม

กำหนดให้ $\vec{x} = \{x_1, x_1, ..., x_N\}$ เป็นโครโมโซมความยาว N, โดยที่แต่ละบิตอาจเป็น 0 หรือ 1, $x_i \in \{0,1\}$ การหาค่าความเหมาะสมของโครโมโซมนี้สามารถคำนวณได้จากสมการที่ 1

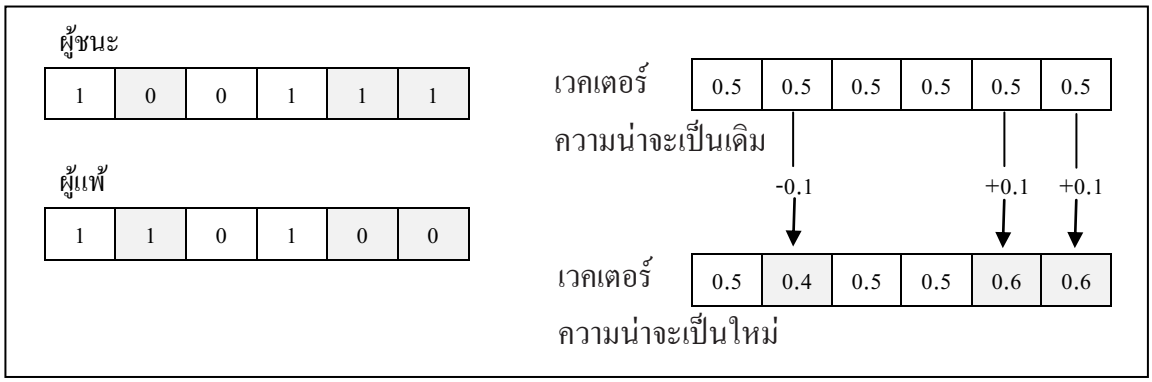
$$F(\vec{x}) = \sum_{i=1}^{N} x_i \qquad (1)$$

การแก้ปัญหานี้โดยใช้ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับสามารถทำได้โดย เริ่มต้นกำหนดเวคเตอร์ความน่าจะเป็นให้ทุกมิติมีค่าความน่าจะเป็นเท่ากับ 0.5 ซึ่งหมายถึงว่าโอกาสที่จะสุ่มตัวอย่างคำตอบออกมาได้แต่ละบิตเป็น 0 หรือ 1 มีโอกาสเท่าๆ กัน

จากนั้น ทำการสุ่มตัวอย่างขึ้นมา 2 ตัว สมมุติสุ่มสร้างคำตอบออกมาได้ดังนี้ คำตอบตัวที่ 1 คือ 1 0 0 1 1 1 คำตอบตัวที่ 2 คือ 1 1 0 1 0 0

คำนวณค่าความเหมาะสมของคำตอบที่สุ่มขึ้นมา ซึ่งในตัวอย่างนี้เป็นปัญหาจำนวนบิตหนึ่งมากสุด ค่าความเหมาะสมคิดจากจำนวนบิตที่เป็น 1 ที่ปรากฏในโครโมโซม ดังนั้นค่าความเหมาะสมของคำตอบตัวที่ 1 คือ 4 และค่าความเหมาะสมของคำตอบตัวที่ 2 คือ 3 เมื่อทราบค่าความเหมาะสมของคำตอบทั้งสองแล้ว จะนำมาเปรียบเทียบกันเพื่อหาผู้ชนะและผู้แพ้ ซึ่งในที่นี้คำตอบตัวที่ 1 คือผู้ชนะ และคำตอบตัวที่ 2 คือผู้แพ้ เมื่อได้ผู้ชนะและผู้แพ้แล้ว เวคเตอร์ความน่าจะเป็นจะถูกปรับปรุงให้เข้าใกล้คำตอบที่เป็นผู้ชนะ โดยการปรับค่าจะพิจารณาจากบิตที่ต่างกันระหว่างผู้ชนะและผู้แพ้ ซึ่งในที่นี้จะพิจารณาบิตที่ 2, 5 และ 6

สำหรับบิตที่ 2 ผู้ชนะเป็นบิต 0 และผู้แพ้เป็นบิต 1 ดังนั้นเวคเตอร์ความน่าจะเป็นในตำแหน่งที่ 2 จะถูกปรับค่าให้เข้าใกล้คำตอบที่ดีกว่า นั่นคือปรับค่าความน่าจะเป็นให้เข้าใกล้ 0 สมมุติว่ากำหนดค่า *n* เป็น 10 (ซึ่งสามารถเทียบได้กับการกำหนดจำนวนประชากรเป็น 10 ในขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย) ค่าในเวคเตอร์ความน่าจะเป็นจะถูกปรับลดลงเท่ากับ 1/*n* นั่นคือ 0.1



ภาพที่ 2 ตัวอย่างการปรับปรุงเวคเตอร์ความน่าจะเป็น

ทำให้ค่าความน่าจะเป็นในตำแหน่งที่ 2 คือ 0.5 – 0.1 = 0.4

สำหรับบิตที่ 5 และ 6 ผู้ชนะเป็นบิต 1 ดังนั้นการปรับค่าเวคเตอร์ความน่าจะเป็นจะถูกปรับขึ้น จึงได้ค่าใหม่คือ 0.5 + 0.1 = 0.6 ดังตัวอย่างที่แสดงในภาพที่ 2

การดำเนินการของขั้นตอนวิธีจะวนซ้ำจนกว่าทุกๆ บิตในเวคเตอร์ความน่าจะเป็นจะลู่เข้าสู่ 0.0 หรือ 1.0 ทั้งหมดจึงหยุดการทำงานซึ่งจากตัวอย่างข้างต้น สำหรับปัญหานี้เวคเตอร์ความน่าจะเป็นที่เป็นผลลัพธ์ควรจะได้เป็น

1.0	1.0	1.0	1.0	1.0	1.0

ซึ่งคือตัวแบบของผลลัพธ์ที่ดีที่สุด

เพื่อเป็นตัวอย่างที่แสดงความใกล้เคียงกันของประสิทธิภาพของขั้นตอนวิธีวิธีเชิงพันธุกรรมอย่างง่ายและแบบกระชับ ผู้เขียนบทความได้เขียนโปรแกรมเพื่อเปรียบเทียบประสิทธิภาพขั้นตอนวิธีทั้งสอง โดยทดสอบกับปัญหาจำนวนบิตหนึ่งมากสุดขนาด 100 บิต

ในการทดลอง ได้ทำการกำหนดจำนวนประชากรของขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายเริ่มต้นที่ประชากร 4 ตัว จนถึง 100 ตัว โดยทดลองเพิ่มประชากรทีละ 8 ตัวตามลำดับ ในขณะที่ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับมีการสุ่มตัวอย่างเพียง 2 ตัว ขั้นตอนวิธีจะทำงานจนลู่เข้าสู่คำตอบแล้วจึงนำผลลัพธ์มาสรุปเป็นกราฟในภาพที่ 3(ก) โดยผลการทดลองแสดงจากค่าเฉลี่ยของการทดสอบโปรแกรมทั้งสิ้น 50 ครั้ง แกนนอนแสดงจำนวนประชากรของขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย (และคือค่า n ที่ใช้ในการปรับค่าของเวคเตอร์ความน่าจะเป็นในขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ) ส่วนแกนตั้งแทนค่าความเหมาะสมของคำตอบ

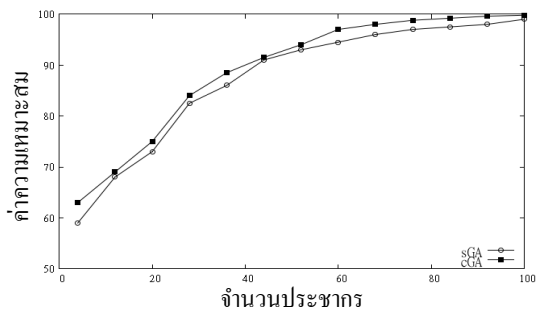
จากภาพที่ 3(ก) จะเห็นได้ว่าขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายและแบบกระชับให้ผลลัพธ์ที่มีคุณภาพใกล้เคียงกันโดยใช้จำนวนครั้งในการประเมินค่าความเหมาะสมเท่ากัน ดังแสดงในภาพที่ 3(ข) ซึ่งผลการทดลองที่ได้สอดคล้องกับ Halik *et al*. (1999) ที่ได้พิสูจน์ไว้ว่าทั้งสองขั้นตอนวิธีมีประสิทธิภาพใกล้เคียงกัน

การประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ

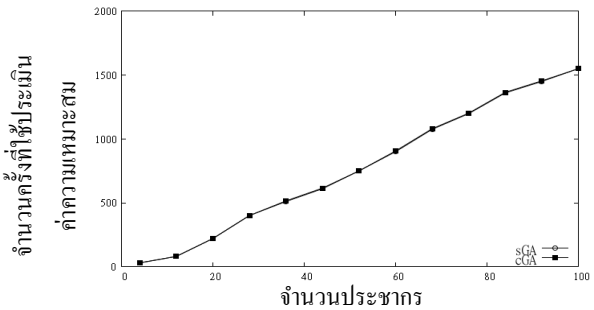
จากหลักการและผลการทดลองของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับที่ได้กล่าวไปข้างต้น จะเห็นได้ว่ากระบวนการทำงานของอัลกอริทึมนี้ไม่ซับซ้อน และยังใช้หน่วยความจำในการประมวลผลน้อยมาก เนื่องจากต้องการเนื้อที่ในการเก็บตัวอย่างคำตอบที่สุ่มมาเพียง 2 ตัวเท่านั้น ในขณะที่ผลลัพธ์การทำงานที่ได้สามารถเทียบเคียงได้กับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่าย

ด้วยความโดดเด่นในจุดนี้ จึงทำให้ขั้นตอนวิธีนี้ถูกนำไปประยุกต์ใช้ในงานหลากหลายด้าน โดยเฉพาะกับงานที่มีข้อจำกัดในเรื่องของหน่วยความจำ และต้องการความรวดเร็วในการประมวลผล ดังตัวอย่างของการนำไปใช้ในฮาร์ดแวร์ ซึ่งแต่เดิม การนำขั้นตอนวิธีเชิงพันธุกรรมไปทำงานในฮาร์ดแวร์เป็นเรื่องยาก เนื่องจากขั้นตอนวิธีเชิงพันธุกรรมต้องอาศัยประชากรจำนวนมากในการหาคำตอบ รวมทั้งต้องการความสามารถของฮาร์ดแวร์สำหรับการประมวลผลค่อนข้างสูง จึงเกิดข้อจำกัดในการนำขั้นตอนวิธีเชิงพันธุกรรมไปใช้งานในอุปกรณ์จริง แต่ด้วยความสามารถของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ ทำให้สามารถนำไปสร้างวงจรฮาร์ดแวร์บน FPGA เพื่อใช้งานได้จริง (Aporntewan & Chongstitvatana, 2001; León-Javier *et al*., 2009)

งานวิจัยส่วนใหญ่มักจะนำขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับไปใช้ในการหาค่าเหมาะสุด (Optimization) และทุกๆ บทความที่ตีพิมพ์ล้วนสรุปว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับสามารถช่วยแก้ปัญหา และปรับปรุงคุณภาพของคำตอบให้ดีขึ้นได้ โดยที่ใช้หน่วยความจำและต้นทุนการคำนวณน้อย ตัวอย่างของงานวิจัยในกลุ่มนี้ เช่น Afshar & Rasa (2006) นำไปใช้ในการค่าหาเหมาะสมสุดของการวางระบบท่อ โดยคำนึงถึงขนาดของท่อและต้นทุนที่ใช้ Cupertino *et al*. (2006) นำไปใช้ในการปรับค่าการควบคุมมอเตอร์ด้วยตนเอง (Self-tuning) Olarthichachart *et al*. (2010) ใช้ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับช่วยในการ



3(ก)



3(ข)

ภาพที่ 3 ผลการทดลองเปรียบเทียบขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายและแบบกระชับ

จัดความถี่ของตารางการเดินรถในมหาวิทยาลัย หรือในงานด้านเครือข่ายคอมพิวเตอร์และการสื่อสาร มีตัวอย่างการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับในการเพิ่มประสิทธิภาพการบริหารจัดการการทำงานของเซ็นเซอร์เน็ตเวิร์ค (Jin *et al*., 2005) และนำไปใช้ในการลดจำนวนทรัพยากรในการสื่อสารผ่านเครือข่าย (Xing & Qu, 2011)

งานวิจัยข้างต้นเป็นการนำขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับไปใช้กับปัญหาที่สามารถแทนคำตอบด้วยสายอักขระเท่านั้น แต่ในปัญหาบางประเภทที่คำตอบเป็นเลขจำนวนจริง การหาคำตอบโดยแทนเลขจำนวนจริงด้วยสายอักขระนั้นให้ผลลัพธ์ไม่ดีเท่าที่ควร Radiom (2007) จึงปรับปรุงขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้เป็นขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับแบบจำนวนจริง (Real-coded compact genetic algorithm) โดยการแทนเวคเตอร์ความน่าจะเป็นด้วยฟังก์ชันการแจกแจงความน่าจะเป็น (Probability distribution function) ของตัวแปรต่างๆ ที่เป็นจำนวนจริง และนำไปประยุกต์ใช้ในการหาค่าเหมาะสุดของการออกแบบขนาดสายอากาศขนาดเล็ก

ถึงแม้ว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับจะมีจุดเด่นในเรื่องของความรวดเร็วในการทำงานและการใช้หน่วยความจำน้อย แต่ก็มีข้อจำกัดในเรื่องความสามารถในการแก้ปัญหาที่ซับซ้อน ดังนั้นในช่วงเวลาที่ผ่านมา จึงมีงานวิจัยจำนวนมากที่เสนอวิธีการปรับปรุงขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับให้มีประสิทธิภาพดีกว่าเดิม ตัวอย่างเช่น Ahn & Ramakrishna (2003) ได้เสนอวิธีการรักษาคำตอบที่ดีเอาไว้ โดยแบ่งเป็น 2 ขั้นตอนวิธี คือ การเก็บคำตอบที่มีค่าความเหมาะสมสูงเอาไว้จนกว่าจะมีคำตอบอื่นที่ดีกว่ามาแทนที่ (Persistent elitist compact genetic algorithm: pe-cGA) และการเก็บคำตอบที่มีค่าเหมาะสมมากสุดในขณะนั้นภายในช่วงระยะเวลาที่กำหนดไว้เพียงระยะเวลาหนึ่งเท่านั้น (Nonpersistent elitist compact genetic algorithm: ne-cGA) ซึ่งทั้ง 2 วิธีการนี้สามารถช่วยหลีกเลี่ยงปัญหาการลู่เข้าสู่คำตอบที่เร็วเกินไป (Premature convergence) ซึ่งมักจะเป็นปัญหาที่ทำให้ขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับสิ้นสุดการทำงานก่อนที่จะเจอคำตอบดีสุดที่แท้จริง (Optimal solution) นอกเหนือจากวิธีการเก็บคำตอบที่มีค่าความเหมาะสมสูงเอาไว้เพื่อเพิ่มประสิทธิภาพของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับแล้ว การปรับปรุงประสิทธิภาพยังทำได้โดยการเปลี่ยนวิธีการปรับเปลี่ยนค่าในเวคเตอร์ความน่าจะเป็น เช่น การใช้ค่าเฉลี่ยเคลื่อนที่ในการชลอการตัดสินใจปรับค่าความน่าจะเป็นออกไป เพื่อให้เห็นแนวโน้มของคำตอบที่ดี (Rimcharoen *et al*., 2006) การเปลี่ยนแปลงวิธีการในการปรับค่าความน่าจะเป็นในเวคเตอร์ยังคงเป็นหัวข้อการวิจัยที่น่าสนใจ เนื่องจากการเปลี่ยนวิธีปรับปรุงค่าของเวคเตอร์เพียงเล็กน้อยนี้ส่งผลให้ขั้นตอนวิธีมีประสิทธิภาพมากขึ้นอย่างเห็นได้ชัด

สรุป

บทความนี้นำเสนอหลักการของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับ การนำไปใช้ ตัวอย่างการประยุกต์ และการปรับปรุงประสิทธิภาพของขั้นตอนวิธีดังกล่าว วัตถุประสงค์ของบทความนี้คือการนำเสนอที่ผู้อ่านจะสามารถเข้าใจได้โดยง่ายและเห็นว่ากระบวนการทำงานของขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับนี้ง่ายและไม่ซับซ้อน เป็นขั้นตอนวิธีที่ใช้หน่วยความจำในการประมวลผลน้อยมาก ในขณะที่ผลลัพธ์การทำงานที่ได้สามารถเทียบเคียงได้กับขั้นตอนวิธีเชิงพันธุกรรมอย่างง่ายที่ใช้กลุ่มประชากรในการหาคำตอบ งานวิจัยส่วนใหญ่จึงมักหยิบยกขั้นตอนวิธีนี้ไปใช้แก้ปัญหาในงานที่มีข้อจำกัดในเรื่องของหน่วยความจำ และต้องการความรวดเร็วในการประมวลผล ซึ่งจากงานวิจัยที่ผ่านมาก็สรุปไปในทิศทางเดียวกันว่าขั้นตอนวิธีเชิงพันธุกรรมแบบกระชับสามารถช่วยปรับปรุงประสิทธิภาพของคำตอบได้เป็นที่น่าพอใจ

เอกสารอ้างอิง

Afshar, A. H., & Rasa, E. (2006). Pipe network optimization using compact genetic algorithm. In *Proceedings of the 7th International Congress on Civil Engineering*.

Ahn, C. W. & Ramakrishna, R. S. (2003). Elitism-based compact genetic algorithm. *IEEE Transactions on Evolutionary Computation, 7*(4), 367-385.

Aporntewan, C. & Chongstitvatana, P. (2001). A hardware implementation of the compact genetic algorithm. In *Proceedings of the 2001 IEEE Congress Evolutionary Computation*.

Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. *Technical Report CMU-CS-95-163*, Carnegie Mellon University.

Cupertino, F., Mininno, E., & Naso, D. (2006). Elitist compact genetic algorithms for induction motor self-tuning control. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*.

Darwin, C. R. (1859). *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. London: John Murray.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1965). Artificial intelligence through a simulation of evolution. In *Biophysics and Cybernetic Systems: Proceedings of the 2nd Cybernetic Sciences Symposium*.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. New York: John Wiley & Sons.

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

Harik, G. R., Lobo, F. G., & Golberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation, 3*(4), 287-297.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press.

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing, 2*, 88-105.

Jin, M-H., Liu, W-Z., Hsu, D. F., & Kao, C-Y. (2005). Compact genetic algorithm for performance improvement in hierarchical sensor networks management. In *Proceedings of the 8th International Symposium on Parallel Architectures, Algorithms and Networks*.

León-Javier, A., Moreno-Armendáriz, M. A., & Cruz-Cortés, N. (2009). Designing a compact genetic algorithm with minimal FPGA resources. *Advances in Intelligent and Soft Computing, 116*, 349-357.

Mühlenbein, H., & Paaß, G. (1996). From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature*. Berlin: Springer.

Olarthichachart, P., Kaitwanidvilai, S., & Karnprachar, S. (2010). Trip frequency scheduling for traffic transportation management based on compact genetic algorithm. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*.

Pelikan, M., Goldberg, D. E., & Lobo, F. G. (1999). *A survey of optimization by building and using probabilistic models*. Urbana, IL: University of Illinois Genetic Algorithms Laboratory (IlliGAL Report No. 99018).

Radiom, S., Aliakbarian, H., Vandenbosch, G., & Gielen, G. (2007). Simple real-coded compact genetic algorithm: application to antenna design. In *Proceedings of the Asia-Pacific Microwave Conference*.

Rechenberg, I. (1965). *Cybernetic solution path of an experimental problem*. Ministry of Aviation, Royal Aircraft Establishment, U.K..

Rechenberg, I. (1971). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog.

Rimcharoen, S., Sutivong, D., & Chongstitvatana, P. (2006). Updating strategy in compact genetic algorithm using moving average approach. In *Proceedings of the 2006 IEEE International Conferences on Cybernetics & Intelligent Systems (CIS) and Robotics, Automation & Mechatronics (RAM)*.

Xing, H., & Qu, R. (2011). A compact genetic algorithm for the network coding based resource minimization problem. *Applied Intelligence*, Online First™, 16 May 2011.