
ขั้นตอนวิธีเชิงพันธุกรรมสำหรับการหาค่าเหมาะที่สุดวัตถุประสงค์หลายอย่าง

Genetic Algorithms for Multi-objective Optimization

กิตติพงษ์ บัญโล่ง*

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา

Kittipong Boonlong*

Department of Mechanical Engineering, Faculty of Engineering, Burapha University

บทคัดย่อ

บทความนี้นำเสนอข้อจำกัดหนึ่งของขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์หลายอย่าง (MOGA) ในการแก้ปัญหาการหาค่าเหมาะที่สุดวัตถุประสงค์หลายอย่างโดยได้นำ MOGA 3 ชนิดคือ NSGA-II, SPEA-II และ COGA-II มาทดลองกับปัญหาวัตถุประสงค์หลายอย่างมาตรฐาน DTLZ2 และ DTLZ6 ที่มีวัตถุประสงค์ 3-6 อย่าง ซึ่งจากผลการทดลองจะพบว่า MOGA ทั้งสามสามารถแก้ปัญหาที่มีวัตถุประสงค์ 3 อย่างได้เป็นอย่างดี COGA-II ดีกว่า NSGA-II และ SPEA2 อย่างชัดเจนสำหรับปัญหาที่มีวัตถุประสงค์ 4 อย่างขึ้นไป นอกจากนี้ NSGA-II และ SPEA2 แก้ปัญหาที่มี 5 วัตถุประสงค์ขึ้นไปไม่ได้ดี ส่วน COGA-II ซึ่งเป็น MOGA ซึ่งพัฒนามาเพื่อแก้ปัญหาดังกล่าวได้แม้ไม่ดีมากดังเช่นปัญหาที่มี 3 วัตถุประสงค์ ซึ่งแสดงถึงข้อจำกัดสำหรับ MOGA ที่มีอยู่ในปัจจุบัน จากผลการทดลองแสดงว่าจำเป็นต้องมีการพัฒนา MOGA เพื่อปัญหาที่มีวัตถุประสงค์จำนวนมากต่อไป

คำสำคัญ : ขั้นตอนวิธีเชิงพันธุกรรม การหาค่าเหมาะที่สุดวัตถุประสงค์หลายอย่าง

Abstract

This paper presents a limitation of multi-objective genetic algorithm (MOGA) for solving multi-objective optimization problems. The paper investigates 3 MOGAs – NSGA-II, SPEA-II and COGA-II to solve benchmark problems – DTLZ2 and DTLZ6 with 3-6 objectives. The results show that all MOGAs can solve the problems with 3 objectives effectively. COGA-II is obviously better than the others for the problems with 4-or-more objectives. In addition, NSGA-II and SPEA-II cannot well solve the problems with 5-or-more objectives. On the other hand, COGA-II can solve these problems although it is not very good as in the problems with 3 objectives. This shows the limitation of current MOGAs. Thus, it is necessary to develop MOGA for the optimization problems with high number of objectives.

Keywords : Genetic Algorithm, Multi-objective Optimization

*Corresponding author. E-mail: kittipong@buu.ac.th

ในปัจจุบันนี้มีวิธีการหาค่าเหมาะที่สุดหลักๆ อยู่ 2 แบบ ซึ่งก็คือวิธีที่ต้องใช้อนุพันธ์ (Derivative-based Method) และวิธีที่ไม่ต้องใช้อนุพันธ์ (Derivative-free Method) แบบที่ใช้อนุพันธ์ตัวอย่างเช่น เกรเดียนท์ของนิวตัน (Newton's Gradient) และเกรเดียนท์สังยุค (Conjugate Gradient) ถูกใช้อย่างแพร่หลายในช่วงทศวรรษที่ 90 ขณะที่ในช่วงเวลาดังกล่าววิธีที่ไม่ต้องใช้อนุพันธ์เช่น ขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm; GA) และขั้นตอนวิธีจำลองอบเหนียว (Simulated Annealing; SA) ถูกใช้ในน้อยในงานทางด้านวิศวกรรม เครื่องกล โดยส่วนใหญ่จะใช้ในงานการควบคุม ปัญญาประดิษฐ์ (Artificial Intelligence) และ CAD/CAM (Masoum *et al.* 2004 และ Marion *et al.* 2006) หากเปรียบเทียบกับวิธีที่ใช้อนุพันธ์แล้ววิธีที่ไม่ใช้อนุพันธ์จะใช้การหาฟังก์ชันวัตถุประสงค์แบบซ้ำๆ และได้มาซึ่งแนวทางการหาค่าตอบภายใต้ทิศทางฮิลิสติกส์ตามหลักธรรมชาติ แม้ว่าโดยทั่วไปวิธีที่ไม่ต้องใช้อนุพันธ์จะหาค่าตอบได้ช้ากว่าวิธีที่ใช้ค่าอนุพันธ์ แต่สำหรับปัญหาที่มีฟังก์ชันวัตถุประสงค์ที่ซับซ้อนและปัญหาแบบสหสาขา วิธีที่ไม่ต้องใช้อนุพันธ์จะมีประสิทธิภาพมากกว่าเพราะวิธีนี้ใช้ได้กับปัญหาที่ไม่สามารถหาอนุพันธ์ได้ โดยเฉพาะอย่างยิ่งปัจจุบันซึ่งใช้คอมพิวเตอร์ในการช่วยคำนวณทำให้ค้นหาค่าตอบได้เร็วขึ้น

GA เป็นวิธีหาค่าตอบเหมาะที่สุดแบบกลุ่มประชากรที่ไม่ใช้อนุพันธ์โดยกลไกในการหาค่าตอบจะอิงกับแนวคิดของดาร์วินในเรื่องการอยู่รอดของผู้ที่แข็งแรงที่สุด เริ่มต้น GA ใช้สำหรับปัญหาการหาค่าเหมาะที่สุดวัตถุประสงค์เดียว (Single-objective Optimization Problem; SOOP) หลังจากนั้นก็มีการมีการดัดแปลงใช้แก้ปัญหาการหาค่าเหมาะที่สุดสำหรับปัญหาที่มีวัตถุประสงค์หลายอย่าง (Multi-objective Optimization Problem; MOOP) ซึ่งจะมีฟังก์ชันวัตถุประสงค์เพื่อทำให้น้อยที่สุดหรือมากที่สุดจำนวนหนึ่ง โดย GA สำหรับการหาค่าเหมาะที่สุดจะเรียกว่า ขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์หลายอย่าง (Multi-objective Genetic Algorithm; MOGA)

ปัญหาการหาค่าเหมาะที่สุดโดยทั่วไปจะเป็นปัญหาวัตถุประสงค์หลายอย่าง ตัวอย่างเช่นการตัดสินใจที่จะซื้อคอมพิวเตอร์โดยพิจารณาจากประสิทธิภาพและราคาก็เป็นตัวอย่างหนึ่งของ MOOP ซึ่งมีสองวัตถุประสงค์ซึ่งก็คือให้ประสิทธิภาพมากที่สุด และให้ราคาน้อยที่สุด เนื่องจากแทบจะเป็นไปไม่ได้เลยที่คำตอบเพียงตัวเดียวจะให้ค่าดีที่สุดทุกวัตถุประสงค์

สำหรับปัญหา MOOP ใด โดยกลุ่มของคำตอบที่ดีที่สุดสำหรับปัญหา MOOP ซึ่งเรียกว่า กลุ่มของผลเฉลยพาเรโตที่แท้จริง (True Pareto Optimal Solutions) สำหรับปัญหาวัตถุประสงค์เดียวนั้นเป็นการง่ายสำหรับเปรียบเทียบว่าคำตอบใดดีกว่า ขณะที่คำตอบของ MOOP นั้นจะเปรียบเทียบโดยใช้หลักการครอบงำแบบพาเรโต (Pareto Domination) ซึ่งจะอธิบายอย่างละเอียดในส่วนของ 2 ของบทความนี้ โดยหากคำตอบหนึ่งครอบงำ (Dominate) อีกคำตอบหนึ่งตามหลักการครอบงำแบบพาเรโต ก็แสดงคำตอบนั้นดีกว่า ดังนั้นสำหรับคำตอบกลุ่มหนึ่ง คำตอบที่ไม่ถูกครอบงำ (Non-dominated Solutions) ก็คือคำตอบที่ดีที่สุดของกลุ่มคำตอบนี้

ขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์หลายอย่าง (MOGA) ใช้หลักการของการครอบงำแบบพาเรโตเข้ากับขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์เดียว (SOGA) สำหรับ SOGA คำว่าวัตถุประสงค์ของคำตอบใดๆ จะถูกใช้โดยตรงในการหาค่าความแข็งแรงของคำตอบนั้น ในอีกด้านหนึ่ง MOGA จะใช้หลักการครอบงำแบบพาเรโตในการให้ค่าของแข็งแรงหรือลำดับชั้น (Rank) ของคำตอบใดๆ จากคำตอบวัตถุประสงค์ทั้งหมดของคำตอบนั้น เป้าหมายของ MOGA ก็คือเพื่อที่จะหาของกลุ่มของคำตอบที่ไม่ถูกครอบงำที่ดี ซึ่งจะต้องมีคุณสมบัติสองอย่างซึ่งก็คือ คำตอบกลุ่มนี้ต้องเข้าใกล้คำตอบที่ดีที่สุดที่แท้จริงซึ่งจะเรียกว่า ผลเฉลยพาเรโตที่แท้จริง (True Pareto Optimal Solutions) และต้องเป็นคำตอบที่มีความหลากหลายด้วย

วิธีการ

บทความนี้จะนำเสนอขั้นตอนวิธีเชิงพันธุกรรมแบบวัตถุประสงค์หลายอย่าง (MOGA) สำหรับปัญหาการหาค่าเหมาะที่สุดวัตถุประสงค์หลายอย่าง โดยเริ่มแรกจะอธิบายขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์เดียว ต่อด้วยการหาค่าเหมาะที่สุดแบบวัตถุประสงค์หลายอย่าง และ MOGA

1. ขั้นตอนวิธีเชิงพันธุกรรมแบบวัตถุประสงค์เดียว

ขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์เดียว (Single-objective Genetic Algorithm, SOGA) ถูกอธิบายอย่างละเอียดใน (Goldberg, 1989) รวมถึงปัจจุบันก็มีงานวิจัยมากมายที่ใช้และพัฒนา SOGA เช่น Kaya (2011) และ Tsai (2011) โดยบทความนี้จะแสดงส่วนพื้นฐานและกลไกการทำงานของ GA กระบวนการทำงานของ SOGA เป็นไปตามขั้นตอนดังนี้

1) สร้างกลุ่มของประชากรของคำตอบเริ่มต้นโดยการสุ่ม โดยคำตอบจะเข้ารหัสซึ่งส่วนใหญ่จะเป็นตัวเลขเช่นเข้ารหัสเป็นเลขฐานสอง เลขจำนวนเต็ม และเลขจำนวนเต็ม

โดยแต่ละคำตอบจะแทนด้วย กลุ่มตัวเลขที่ต่อกัน ซึ่งจะเรียกว่า โครโมโซม

2) ทำการถอดรหัสของโครโมโซมทั้งหมดในกลุ่มประชากร เพื่อที่จะได้กลุ่มคำตอบของปัญหาที่พิจารณา

3) หลังจากได้คำตอบที่จากการถอดรหัส ทำการหาค่า ฟังก์ชันวัตถุประสงค์ของแต่ละคำตอบในกลุ่มประชากรทั้งหมด

4) ทำการคำนวณค่าความแข็งแรงของแต่ละคำตอบโดยใช้ ค่าวัตถุประสงค์ที่ได้มา

5) เลือกกลุ่มประชากรรุ่นพ่อแม่จากกลุ่มประชากรรุ่น ปัจจุบัน โดยในการเลือกนั้นจะใช้ค่าความแข็งแรงเป็นตัวบ่งชี้ว่า จะเลือกคำตอบใดจากประชากรปัจจุบันนี้

6) ทำการเปลี่ยนแปลงกลุ่มประชากรรุ่นพ่อแม่ด้วยใช้ ตัวดำเนินการพันธุกรรม ซึ่งก็คือ การสลับสายพันธุ์ (Crossover) และการกลายพันธุ์ (Mutation)

7) กลับไปทำซ้ำขั้นตอนที่ 2 จนกระทั่งคำตอบเข้าสู่หรือ จนกระทั่งครบจำนวนการทำซ้ำที่กำหนด โดย 1 รอบของขั้นตอนที่ 2 ถึง 6 ก็คือ 1 รุ่น (Generation) ของการหาคำตอบด้วยขั้นตอนวิธี เชิงพันธุกรรม คำตอบที่ดีที่สุดของประชากรรุ่นสุดท้ายก็คือผลลัพธ์ ของ GA นอกจากนั้นอาจจะมีการเพิ่มตัวดำเนินการที่เรียกว่า อภิชนนิยม (Elitism) หลังจากขั้นตอนที่ 4) จะมีกลุ่มคำตอบที่ดีๆ เพื่อที่จะเป็นกลุ่มของคำตอบใหม่ของรุ่นถัดไป

2. การหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่าง

ปัญหาการหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่าง และวิธีสำหรับการหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่าง เป็นดังต่อไปนี้

1) ปัญหาการหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่าง

ปัญหาการหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่าง (Multi-objective Optimization Problem, MOOP) จะมี วัตถุประสงค์ซึ่งอาจจะต้องการให้มีค่าน้อยสุดหรือมากที่สุด โดย MOOP อธิบายอยู่ในรูปแบบทั่วไปได้ดังสมการดังต่อไปนี้

$$\begin{aligned} &\text{Minimize/Maximize } f_m(x), m = 1, 2, \dots, M \\ &\text{subject to } g_j(x) = 0, j = 1, 2, \dots, J \\ &\text{and } h_k(x) \geq \text{or } \leq 0, k = 1, 2, \dots, K \end{aligned} \quad (1)$$

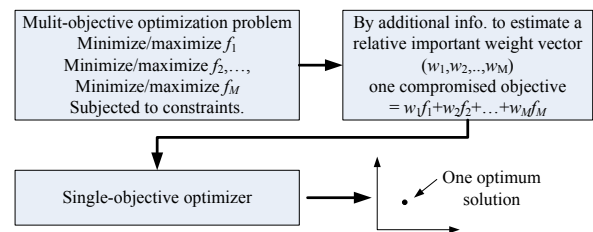
คำตอบ x เป็นเวกเตอร์ซึ่งประกอบด้วย จำนวนตัวแปรตัดสินใจ n ตัว จะได้ว่า $x = \{x_1, x_2, \dots, x_n\}$ โดยมี M วัตถุประสงค์ แทนด้วยเวกเตอร์ $f = \{f_1(x), f_2(x), \dots, f_M(x)\}$ โดยแต่ละวัตถุประสงค์ ต้องการให้เป็นค่าน้อยที่สุดหรือมากที่สุด สำหรับกลุ่มของข้อบังคับแบบเท่ากัน J เงื่อนไข และข้อบังคับแบบไม่เท่ากัน K เงื่อนไข โดย $g_j(x)$ และ $h_k(x)$ เป็นฟังก์ชันเงื่อนไขแบบเท่ากันและไม่เท่ากัน

ตามลำดับ โดยขอบเขตของตัวแปร แต่ละตัวแปรตัดสินใจ x_i จะต้องมีค่าอยู่ระหว่าง ขอบล่าง x_i^l และขอบบน x_i^u

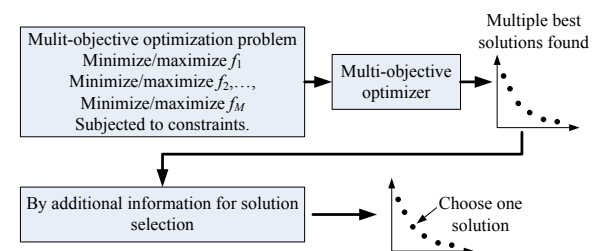
2) วิธีสำหรับการหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่าง

สำหรับการหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่างจะมี 2 วิธีหลักๆ ที่ใช้ ซึ่งก็คือ วิธีผลรวมถ่วงน้ำหนัก (Weighted-sum Approach) และวิธีพื้นฐานพาเรโต (Pareto-based Approach) โดยกระบวนการของวิธีทั้งสองจะอธิบายดังภาพที่ 1 และภาพที่ 2 ตามลำดับ

สำหรับวิธีผลรวมถ่วงน้ำหนักซึ่งเป็นวิธีหาค่าเหมาะที่สุดวัตถุประสงคหลายอย่างแบบดั้งเดิม ทุกวัตถุประสงค์จะจับมารวมกันโดยถ่วงน้ำหนักจนเหลือเพียงวัตถุประสงค์เดียว หลังจากนั้น ก็ใช้ตัวหาค่าเหมาะที่สุดวัตถุประสงค์เดียวในการหาคำตอบ ซึ่งโดยส่วนใหญ่ก็จะได้คำตอบเหมาะที่สุดเพียงคำตอบเดียว แต่ก็จะเป็นคำตอบที่ขึ้นอยู่ความคิดส่วนตัวของผู้ใช้เนื่องจากการกำหนด ค่าถ่วงน้ำหนัก แม้ว่าจะมีคำตอบแบบกลางๆ (Compromised Solution) ดีที่สุดเพียงคำตอบเดียว เทคนิคนี้ก็ค่อนข้างเป็นที่นิยม เนื่องจากง่ายในการใช้



ภาพที่ 1 วิธีผลรวมถ่วงน้ำหนัก



ภาพที่ 2 วิธีพื้นฐานพาเรโต

เนื่องจากเป็นไปไม่ได้ที่จะมีเพียงคำตอบเดียวที่ให้วัตถุประสงค์ดีที่สุดพร้อมกันทุกวัตถุประสงค์สำหรับปัญหาวัตถุประสงค์หลายอย่าง กลุ่มคำตอบที่เหมาะสมที่สุดของปัญหานี้คือคำตอบที่ไม่ถูกครอบงำเมื่อเปรียบเทียบคำตอบทั้งหมดในปริภูมิค้นหา หรือผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริง (True Pareto Optimal Solutions) ซึ่งคำตอบนี้สามารถหาได้โดยวิธีแบบพาเรโตซึ่งเป็น

วิธีแบบที่ 2 ซึ่งใช้หลักการของการครอบงำแบบพาเรโต (Pareto Domination) ซึ่งเป็นดั่งนิยามที่ 1

นิยามที่ 1 การครอบงำแบบพาเรโต

คำตอบ x ใดๆ จะครอบงำหรือดีกว่าคำตอบ y ซึ่งแทนด้วยสัญลักษณ์ $x \prec y$ ถ้าเงื่อนไขทั้งสองข้อต่อไปนี้เป็นความจริง

- ไม่มีวัตถุประสงค์ใด ที่คำตอบ x แย่กว่าคำตอบ y
- มีอย่างน้อย 1 วัตถุประสงค์ ที่คำตอบ x ดีกว่าคำตอบ y

ในการเปรียบเทียบคำตอบของปัญหาการหาค่าเหมาะที่สุด วัตถุประสงค์หลายอย่าง (Multi-objective Optimization Problem, MOOP) สำหรับวิธีนี้สุดท้ายจะได้คำตอบที่ดีที่สุดหลายคำตอบ หลังจากนั้นด้วยการพิจารณาเพิ่มเติม คำตอบที่เหมาะสมหนึ่งตัวจะถูกเลือกมาสำหรับปัญหาดังกล่าว ดังภาพที่ 2

หัวข้อถัดไปจะอธิบายขั้นตอนวิธีเชิงพันธุกรรม สำหรับปัญหา วัตถุประสงค์หลายอย่าง ซึ่งจะเรียกว่า “ขั้นตอนวิธีเชิงพันธุกรรม วัตถุประสงค์หลายอย่าง (Multi-objective Genetic Algorithm, MOGA)”

3. ขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์หลายอย่าง (Multi-objective Genetic Algorithm)

ตอนเริ่มแรกนั้นขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm, GA) สร้างขึ้นมาสำหรับปัญหาที่มีวัตถุประสงค์เดียว หลังจากนั้นก็มีความพยายามช่วงแรกที่จะใช้ GA สำหรับปัญหา วัตถุประสงค์หลายอย่างซึ่งได้แก่การใช้ VEGA (Vector Evaluated Genetic Algorithm) ซึ่งนำเสนอโดย (Schaffer, 1985) โดย VEGA วิเคราะห์รวมถ่วงน้ำหนัก อย่างไรก็ตามพบว่าคำตอบที่ได้จาก VEGA มี วัตถุประสงค์ส่วนหนึ่งเบี่ยงเบนไปทางค่าขอบของวัตถุประสงค์นั้นๆ ซึ่งเป็นข้อเสียของวิธีผลรวมถ่วงน้ำหนัก โดยเฉพาะสำหรับปัญหา หาค่าเหมาะที่สุดวัตถุประสงค์หลายอย่างแบบไม่นูน (Non-Convex Multi-objective Optimization Problem) (Deb, 2001) หลังจากนั้นก็มีการพัฒนาขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์หลายอย่างซึ่งใช้วิธีพื้นฐานพาเรโต ซึ่งไม่มีข้อเสียดังกล่าว โดยใช้ การครอบงำแบบพาเรโตในการจัดลำดับชั้นของคำตอบซึ่งจะเพิ่มเข้าไปในขั้นตอนที่ 4) ของกระบวนการ SOGA โดยเริ่มแรก MOGA ที่ใช้วิธีแบบพาเรโต จะเป็น MOGA แบบไม่เอลิทิสทึม (Non-Elitist MOGA) ได้แก่ FFGA (Fonseca & Fleming, 1993), NPGA (Horn & Nafpliotis, 1993), และ NSGA (Srinivas & Deb, 1994) ขณะที่แบบสมัยใหม่ซึ่งส่วนใหญ่จะเป็น MOGA แบบเอลิทิสทึม (Elitist MOGA) ได้แก่ SPEA (Zitzler & Thiele, 1999), NSGA-II (Deb *et al.*, 2002), SPEA-II (Zitzler *et al.*, 2002), COGA (Maneeratana *et al.*, 2006) และ COGA-II (Boonlong *et al.*,

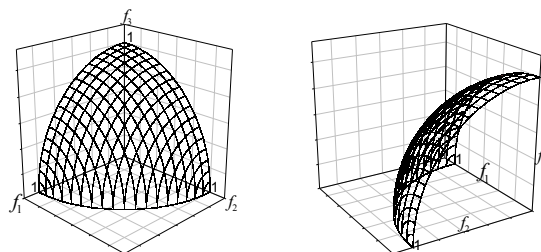
2010)

บทความนี้จะนำเสนอถึงการใช้ MOGA ในการแก้ปัญหา การหาค่าเหมาะที่สุดที่มีวัตถุประสงค์จำนวนมาก โดยจะเลือก ปัญหามาตรฐานที่นิยมซึ่งก็คือปัญหาของ Deb, Thiele, Laumanns, และ Zitzler ซึ่งจะเรียกสั้นๆ ว่า ปัญหา DTLZ (Deb *et al.*, 2005) ซึ่งมีทั้งหมด 7 ปัญหาซึ่งก็คือ DTLZ1-7 ซึ่งทุกปัญหาสามารถปรับเปลี่ยนจำนวนวัตถุประสงค์ได้ตั้งแต่ 2 วัตถุประสงค์ ขึ้นไป โดยบทความนี้จะเลือก DTLZ2 และ DTLZ6 ทั้งนี้ก็เนื่องจาก DTLZ2 เป็นปัญหาที่ง่ายที่สุดที่ไม่มีความยุ่งยากมากนักซึ่งสามารถ ทดสอบประสิทธิภาพข้อจำกัดของการครอบงำแบบพาเรโตสำหรับ ปัญหาที่มีวัตถุประสงค์จำนวนมากได้ ส่วน DTLZ6 ถือเป็นปัญหาที่ ยากมากซึ่งใช้เพื่อทดสอบประสิทธิภาพของ MOGA ที่ใช้ โดยจะ เลือกจำนวนวัตถุประสงค์เท่ากับ 3-6 โดยจะมี MOGA ทั้งหมด 3 ตัว ซึ่งก็คือ NSGA-II, SPEA-II และ COGA-II โดย NSGA-II และ SPEA-II เป็น MOGA ที่ได้รับการยอมรับและเป็นที่นิยม ขณะที่ MOGA ตัวสุดท้าย COGA-II เป็นขั้นตอนวิธีที่พัฒนามาเพื่อปัญหา ที่มีวัตถุประสงค์จำนวนมาก

ผลการทดลองและวิจารณ์ผล

ปัญหา DTLZ2 แสดงดังสมการที่ (2) ซึ่งมีลักษณะโค้งของ คำตอบเหมาะที่สุดที่แท้จริงแบบพื้นผิววงกลม ดังแสดงในภาพที่ 3

$$\begin{aligned} f_1(x) &= (1 + g(\mathbf{x}_m)) \cos(x_1\pi/2) \cdots \cos(x_{m-2}\pi/2) \cos(x_{m-1}\pi/2), \\ f_2(x) &= (1 + g(\mathbf{x}_m)) \cos(x_1\pi/2) \cdots \cos(x_{m-2}\pi/2) \sin(x_{m-1}\pi/2), \\ &\vdots \\ f_m(x) &= (1 + g(\mathbf{x}_m)) \sin(x_1\pi/2) \\ g(\mathbf{x}_m) &= \sum_{x_i \in \mathbf{x}_m} (x_i - 0.5)^2 \end{aligned} \quad (2)$$



ก) ด้านหน้า

ข) ด้านข้าง

ภาพที่ 3 กราฟแสดงด้านหน้าและด้านข้างของคำตอบแท้จริงของ DTLZ2 ที่มี 3 วัตถุประสงค์

จากสมการ $x_i \in [0,1]$ สำหรับ i ใดๆ โดยจะได้คำตอบที่ g เป็นผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริง เมื่อ $g(\mathbf{x}_m) = 0$ ซึ่งจะเป็นจริงเมื่อ $x_i = 0.5$ สำหรับทุก $x_i \in \mathbf{x}_m$ ซึ่งวัตถุประสงค์ f_1 ถึง f_m จะมีความสัมพันธ์เหมือนกับพื้นผิววงกลม ซึ่งจะได้ว่า $\sum_{i=1}^m f_i^2 = 1$

ส่วนปัญหา DTLZ6 นั้น ลักษณะโค้งของคำตอบเหมาะที่สุดที่แท้จริงจะเป็นเส้นโค้งดังภาพที่ 4 โดยปัญหานี้จะเป็นดั่งสมการที่ (3) ปัญหานี้จะยากกว่า DTLZ2 เนื่องจากฟังก์ชัน $g(\mathbf{x}_m)$ ที่ใช้ซึ่งจะทำให้เกิดคำตอบพาเรโตเหมาะที่สุดเฉพาะที่ (Local Pareto Optimal Solution) หลายกลุ่ม ซึ่งจะทำให้การค้นหาคำตอบเป็นไปได้โดยยาก รวมถึงว่าปัญหานี้จะมีการเชื่อมโยงกัน (Coupling) ของตัวแปร ซึ่งหากเปรียบเทียบสมการของ DTLZ2 และ DTLZ6 จะเห็นว่า DTLZ2 ใช้ตัวแปร x_i โดยตรง ขณะที่ DTLZ6 ใช้ตัวแปร θ_i ซึ่งเป็นฟังก์ชันของ x_i และ \mathbf{x}_m ซึ่งมีแสดงการเชื่อมกับของตัวแปร x_i และ \mathbf{x}_m

$$f_1(\mathbf{x}) = (1 + g(\mathbf{x}_m) \cos(\theta_1) \cdots \cos(\theta_{m-2}) \cos(\theta_{m-1})),$$

$$f_2(\mathbf{x}) = (1 + g(\mathbf{x}_m) \cos(\theta_1) \cdots \cos(\theta_{m-2}) \sin(\theta_{m-1})),$$

⋮

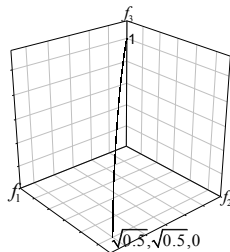
$$f_m(\mathbf{x}) = (1 + g(\mathbf{x}_m) \sin(\theta_1))$$

$$g(\mathbf{x}_m) = \sum_{x_i \in \mathbf{x}_m} x_i^{0.1}$$

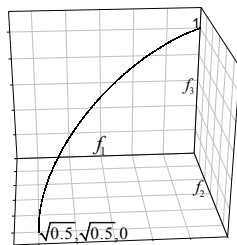
$$\theta_i = x_i \pi / 2$$

$$\theta_i = \frac{\pi}{4(1 + g(\mathbf{x}_m))} (1 + 2g(\mathbf{x}_m)x_i) \text{ for } i = 2, \dots, m-1 \quad (3)$$

จากสมการ $x_i \in [0,1]$ สำหรับ i ใดๆ โดยจะได้คำตอบที่เหมาะสมที่สุดที่แท้จริงเมื่อ $g(\mathbf{x}_m) = 0$ เมื่อ $x_i = 0$ สำหรับทุก $x_i \in \mathbf{x}_m$ ซึ่งเนื่องจากฟังก์ชัน g เป็นผลรวมของ x_i ยกกำลัง 0.1 หรือรากที่ 10 ของ x_i ทำให้ความหนาแน่นของคำตอบต่ำบริเวณใกล้ผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริง ซึ่งเป็นการยากที่จะได้คำตอบที่ใกล้เคียงกับผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริง



ก) ด้านหน้า



ข) ด้านข้าง

ภาพที่ 4 กราฟแสดงด้านหน้าและด้านข้างของคำตอบแท้จริงของ DTLZ6 ที่มี 3 วัตถุประสงค์

ผลการทดสอบโดย Boonlong *et al.*, 2010 ซึ่งใช้ MOGA 3 ตัว ซึ่งก็คือ NSGA-II, SPEA-II, และ COGA-II โดยในการทดลองแต่ละครั้งจะกำหนดให้กลุ่มประชากรมี 100 คำตอบ และให้จำนวนรุ่น (Number of Generation) เท่ากับ 800 ดังนั้นก็จะมีจำนวนครั้งการคำนวณฟังก์ชันวัตถุประสงค์เท่ากับ $100 \times 800 = 80,000$ โดยในแต่ละปัญหาจะมีการดำเนินการซ้ำ 30 ครั้ง ทั้งนี้เนื่องจากในทางปฏิบัติกระบวนการภายในนอกเหนือจากการคำนวณฟังก์ชันวัตถุประสงค์ MOGA ทั้งสามใช้ เวลาร่น้อยมาก จึงไม่จำเป็นต้องเปรียบเทียบเวลาที่ใช้ในการหาคำตอบของแต่ละ MOGA เช่นปัญหา ที่ต้องใช้ระเบียบวิธีเชิงตัวเลขในการคำนวณ เวลาในการหาคำตอบแทบทั้งหมดจะใช้ในการคำนวณฟังก์ชันวัตถุประสงค์ ซึ่งหากมีจำนวนครั้งของการคำนวณดังกล่าวเท่ากันก็ถือได้ว่าใช้เวลาใกล้เคียงกัน

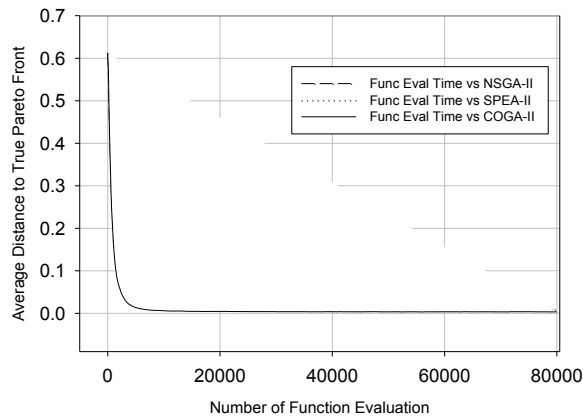
ตารางที่ 1 ระยะห่างเฉลี่ยของคำตอบที่ได้กับผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริงของปัญหา DTLZ2 (Boonlong *et al.*, 2010)

MOGAs	3obj	4obj	5obj	6obj
NSGA-II	0.0090	<u>0.0395</u>	<u>0.4600</u>	<u>1.6093</u>
SPEA-II	<u>0.0089</u>	0.0687	1.3523	2.2346
COGA-II	0.0033	0.0043	0.0108	0.0237

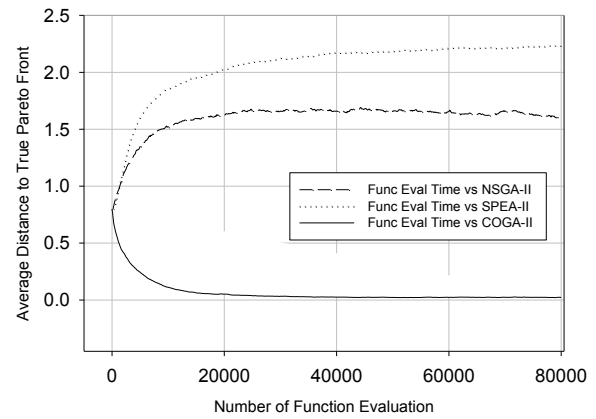
ตารางที่ 2 ระยะห่างเฉลี่ยของคำตอบที่ได้กับผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริงของปัญหา DTLZ6 (Boonlong *et al.*, 2010)

MOGAs	3obj	4obj	5obj	6obj
NSGA-II	<u>0.0641</u>	10.1660	<u>14.3700</u>	<u>19.4510</u>
SPEA-II	0.1269	<u>7.2514</u>	15.2870	20.3620
COGA-II	0.0369	4.6022	6.5050	10.4950

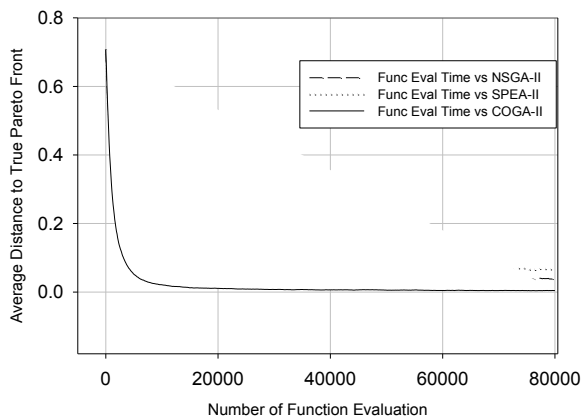
ระยะห่างเฉลี่ยของคำตอบที่ได้กับผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริงของ MOGA ทั้งหมด สำหรับปัญหา DTLZ2 และ DTLZ6 แสดงดังตารางที่ 1 และ ตารางที่ 2 ตามลำดับ รวมถึงมีการแสดงผลโดยกราฟเส้นซึ่งแสดงระยะห่างเฉลี่ยของคำตอบที่ได้กับผลเฉลยพาเรโตเหมาะที่สุดที่แท้จริง (Average Distance to True Pareto Front) บนแกน y ต่อจำนวนการคำนวณฟังก์ชันบนแกน x ซึ่งแสดงดังภาพที่ 5 – ภาพที่ 12



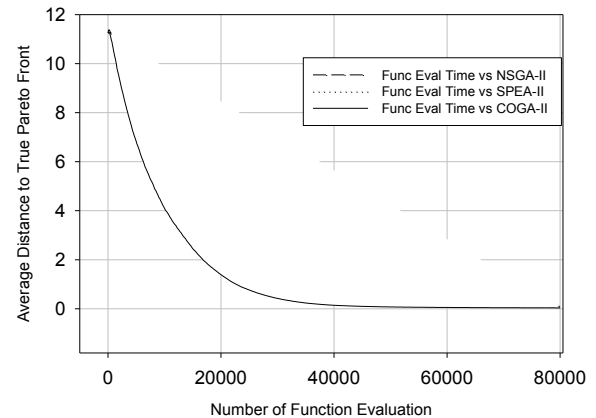
ภาพที่ 5 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ2 ที่มี 3 วัตถุประสงค์ (Boonlong et al., 2010)



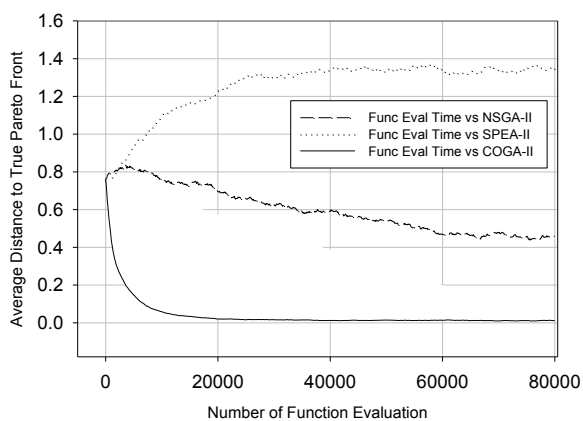
ภาพที่ 8 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ2 ที่มี 6 วัตถุประสงค์ (Boonlong et al., 2010)



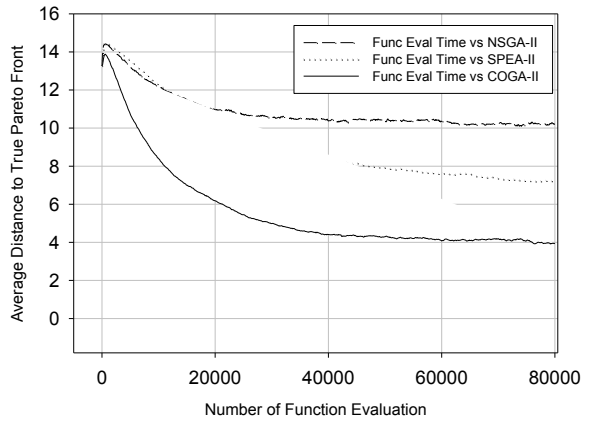
ภาพที่ 6 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ2 ที่มี 4 วัตถุประสงค์ (Boonlong et al., 2010)



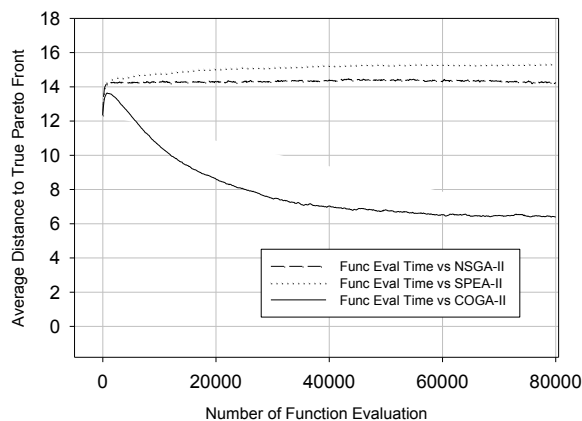
ภาพที่ 9 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ6 ที่มี 3 วัตถุประสงค์ (Boonlong et al., 2010)



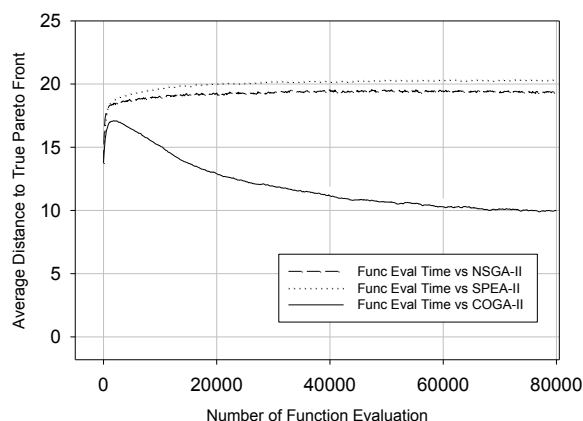
ภาพที่ 7 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ2 ที่มี 5 วัตถุประสงค์ (Boonlong et al., 2010)



ภาพที่ 10 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ6 ที่มี 4 วัตถุประสงค์ (Boonlong et al., 2010)



ภาพที่ 11 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ6 ที่มี 5 วัตถุประสงค์ (Boonlong et al., 2010)



ภาพที่ 12 กราฟแสดงระยะห่างเฉลี่ยของคำตอบของ DTLZ6 ที่มี 6 วัตถุประสงค์ (Boonlong et al., 2010)

จากผลการทดลองซึ่งแสดงดังตารางที่ 1 และตารางที่ 2 COGA-II ซึ่งเป็นขั้นตอนวิธีเชิงพันธุกรรมที่พัฒนาขึ้นมาสำหรับปัญหาที่มีหลายวัตถุประสงค์ ให้ผลการทดลองที่ดีที่สุด แต่ก็พบว่าเมื่อจำนวนวัตถุประสงค์มากขึ้น ระยะห่างเฉลี่ยของคำตอบที่ได้กับคำตอบที่แท้จริง (Average Distance to True Pareto Front) จะมากขึ้น (แย่ลง) แม้ว่าจะเป็นปัญหามulti-objective หลายรูปแบบเดียวกันก็ตาม โดยเฉพาะอย่างยิ่งเมื่อจำนวนวัตถุประสงค์เพิ่มจาก 4 เป็น 5 แม้แต่ในกรณีของ COGA-II ก็ตาม ซึ่งเมื่อเปรียบเทียบผลของ COGA-II สำหรับปัญหา DTLZ2 ในตารางที่ 1 และผลของ COGA-II สำหรับปัญหา DTLZ6 ในตารางที่ 2 ก็จะพบว่า COGA-II ให้ผลที่ค่อนข้างดีสำหรับ DTLZ2 ซึ่งเป็นปัญหาที่ไม่ยากมากนัก โดย COGA-II ให้ระยะห่างเฉลี่ยของ

คำตอบที่ได้กับคำตอบที่แท้จริง ใกล้เคียงกันทุกจำนวนวัตถุประสงค์ สำหรับปัญหา DTLZ2 แต่สำหรับปัญหา DTLZ6 พบว่า COGA-II ให้ผลที่แย่ลงอย่างชัดเจน

หากพิจารณากราฟดังภาพที่ 5 ถึง ภาพที่ 12 ซึ่งแสดงระยะห่างเฉลี่ยของคำตอบที่ได้กับคำตอบที่แท้จริง (Average Distance to True Pareto Front) ต่อจำนวนการคำนวณฟังก์ชัน (Number of Function Evaluation) พบว่าคำตอบจะลู่เข้าหาคำตอบที่แท้จริงได้อย่างรวดเร็วสำหรับปัญหาที่มีเพียงแค่ 3 วัตถุประสงค์ ทั้งสองปัญหา แม้ว่าปัญหา DTLZ6 ที่มี 3 วัตถุประสงค์ซึ่งมีความยากกว่า จะลู่เข้าช้ากว่าก็ตาม แต่เมื่อจำนวนวัตถุประสงค์เป็น 4-6 ก็จะพบว่า COGA-II ให้การลู่เข้าสู่คำตอบที่แท้จริงได้ค่อนข้างดี สำหรับปัญหา DTLZ2 ซึ่งเมื่อพิจารณาจากกราฟสำหรับ DTLZ2 ที่มี 4-6 วัตถุประสงค์ ดังภาพที่ 10-12 ก็จะพบว่า COGA-II ให้จำนวนรอบที่คำตอบลู่เข้าสู่คำตอบที่แท้จริงใกล้เคียงกันทุกวัตถุประสงค์ แต่สำหรับปัญหา DTLZ6 ก็พบว่าการลู่เข้าจะแตกต่างกันยิ่งจำนวนวัตถุประสงค์มากขึ้นคำตอบก็ยิ่งห่างจากคำตอบที่แท้จริงมากขึ้น หากพิจารณาผลที่ได้จาก NSGA-II และ SPEA-II ก็จะพบผลที่คาดไม่ถึง โดยเฉพาะปัญหาที่มี 5-6 วัตถุประสงค์ก็จะพบว่าจำนวนครั้งของการคำนวณฟังก์ชันมากขึ้น แต่กลับคำตอบที่ได้ยิ่งห่างจากคำตอบที่แท้จริงออกไป กลับการเป็นว่าคำตอบลู่ออกจากคำตอบที่แท้จริง สุดท้ายคำตอบที่ได้กลับแย่กว่าคำตอบเริ่มแรกซึ่งได้จากการสุ่ม ซึ่งแสดงถึงจุดอ่อนหนึ่งของสำหรับ MOGA

สรุป

สำหรับปัญหาที่เหมาะสมที่มี 2-3 วัตถุประสงค์นั้น MOGA สามารถแก้ปัญหาได้เป็นอย่างดี แต่จำเป็นต้องมีการพัฒนาขั้นตอนวิธีเชิงพันธุกรรมวัตถุประสงค์หลายอย่าง (Multi-objective Genetic Algorithm; MOGA) สำหรับปัญหาที่มีจำนวนวัตถุประสงค์จำนวนมาก แต่สำหรับนักวิจัยที่เป็นผู้ใช้ MOGA ในหลักการใช้นั้น ควรจะรู้ถึงขีดจำกัดของขั้นตอนวิธีนี้ ซึ่ง MOGA ปัจจุบันสามารถแก้ปัญหาที่มีเพียงแค่ 2-3 วัตถุประสงค์ได้เป็นอย่างดี หรือหากจำเป็นที่จะใช้กับปัญหาที่มีจำนวนวัตถุประสงค์มาก ก็ควรจะเลือกใช้ MOGA ที่พัฒนาขึ้นมาสำหรับปัญหาที่มีวัตถุประสงค์จำนวนมาก เช่น COGA-II หรืออาจจะใช้โดยการลดวัตถุประสงค์ที่จะทำการหาค่าเหมาะที่สุดลงเหลือเพียงแค่ 2-3 วัตถุประสงค์ โดยกำหนดให้วัตถุประสงค์ที่เหลือเป็นเงื่อนไขบังคับ (Constraint) เป็นต้น หรืออาจประยุกต์วิธีถ่วงน้ำหนักพร้อมด้วยเช่น (Soylu, & Köksala, 2010)

- Boonlong, K. (2010) Improved Compressed Objective Genetic Algorithm (COGA-II). In *Proceeding of the International Conference on Evolutionary Computation*,(pp.95-103), Valencia, Spain.
- Deb, K. (2001) *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.
- Deb, K.; Agrawal, S.; Pratap, A.; and Meyarivan T. (2002) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6(2): 182-197.
- Deb, K.; Thiele, L.; Laumanns, M; and Zitzler, E. (2005) Scalable test problems for evolutionary multi-objective optimization. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. Berlin, Germany, Springer.
- Fonseca, C. M.; and Fleming, P. J. (1993) Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416-423. Urbana-Champaign, IL, USA.
- Goldberg, D. E. (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley.
- Horn, J.; Nafpliotis, N.; and Goldberg, D. E. (1994) Multi-objective optimization using the niched Pareto genetic algorithm. *First IEEE conference on Evolutionary Computations, IEEE Word Congress on Computational Intelligence, Volume 1*, 82-87.
- Kaya, M. (2011) The effects of two new crossover operators on genetic algorithm performance. *Applied Soft Computing*, 11(1): 881-890.
- Maneeratana, K.; Boonlong, K.; and Chaiyaratana, N. (2006) Compressed-objective genetic algorithm. *Lecture Notes in Computer Science*. 4193: 473-482.
- Marion, R.; Michel, R.; and Faye, C. (2006) Atmospheric correction of hyperspectral data over dark surfaces via simulated annealing. *IEEE Transactions on Geoscience and Remote Sensing*. 44(6): 1566-1574.
- Masoum, M. A. S.; Ladjevardi, M.; Jafarian, A.; and Fuchs, E. F. (2004) Optimal placement, replacement and sizing of capacitor banks in distorted distribution networks by genetic algorithms. *IEEE Transactions on Power Delivery*. 19(4): 1794-1801.
- Schaffer, J. D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. *Proceeding of the First International Conference on Genetic Algorithms*, 93-100, Mahwah, NJ, USA.
- Soylu, B. and Köksala, M. (2010) A favorable weight-based evolutionary algorithm for multiple criteria problems, *IEEE Transactions on Evolutionary Computation*, 14(2) 191-205.
- Srinivas, N.; and Deb, K. (1994) Multi-objective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*, 2(3): 221-248.
- Tsai, H.C.; (2011) Weighted operation structures to program strengths of concrete-typed specimens using genetic algorithm. *Expert Systems with Applications*, 38(1): 161-168.
- Zitzler, E.; Laumanns, M.; and Thiele, L. (2002) SPEA-II: Improving the strength Pareto evolutionary algorithm. *TIK-Report No. 103*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.
- Zitzler, E.; and Thiele, L. (1999) Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*. 3(4): 257-271.