



รายงานวิจัยฉบับสมบูรณ์

โครงการ

การเพิ่มสมรรถนะของขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมาย
สำหรับปัญหาการออกแบบที่ใช้เวลามากในการหาค่าฟังก์ชัน

โดย

สุจินต์ บุรีรัตน์

29 กุมภาพันธ์ 2555

รายงานวิจัยฉบับสมบูรณ์

โครงการ

การเพิ่มสมรรถนะของขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมาย
สำหรับปัญหาการออกแบบที่ใช้เวลามากในการหาค่าฟังก์ชัน

ผู้วิจัย

นาย สุจินต์ บุรีรัตน์

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

มหาวิทยาลัยขอนแก่น

สนับสนุนโดย

สำนักงานคณะกรรมการการอุดมศึกษาและสำนักงานกองทุนสนับสนุนการวิจัย

(ความเห็นในรายงานนี้เป็นของผู้วิจัย สกอ. และ สกว. ไม่จำเป็นต้องเห็นด้วยเสมอไป)

กิตติกรรมประกาศ

ผู้วิจัยขอขอบคุณ สำนักงานคณะกรรมการการอุดมศึกษาและสำนักงานกองทุนสนับสนุนการวิจัย สำหรับทุนสนับสนุนการวิจัย และขอขอบคุณหน่วยงานต้นสังกัดสำหรับการส่งเสริมให้กิจกรรมนี้ดำเนินไปได้ด้วยดี ขอขอบคุณนักศึกษาบัณฑิตศึกษาที่ทำหน้าที่ผู้ช่วยวิจัยที่ดียิ่ง ท้ายนี้ขอขอบคุณผู้ใกล้ชิดทั้งหลายที่คอยให้กำลังใจ

บทคัดย่อ

รหัสโครงการ: RSA5280006

ชื่อโครงการ: การเพิ่มสมรรถนะของขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมายสำหรับปัญหาการออกแบบที่ใช้เวลามากในการหาค่าฟังก์ชัน

ชื่อนักวิจัย และสถาบัน: นายสุจินต์ บุรีรัตน์ ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

อีเมล: sujbur@kku.ac.th

ระยะเวลาโครงการ 3 ปี

งานวิจัยนี้ศึกษาแนวทางการเพิ่มสมรรถนะในการหาผลเฉลยของขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมายสำหรับการออกแบบ หาค่าเหมาะที่สุดแบบหลายเป้าหมายของระบบทางวิศวกรรมที่มีการใช้งานจริง แนวทางเบื้องต้นของการวิจัยคือ การนำเอาขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมายที่มีอยู่แล้ว และได้รับการพิสูจน์แล้วว่ามีความสมรรถนะในการหาผลเฉลยเหมาะที่สุด เป็นวิธีหลัก จากนั้นทำการผสมผสานแบบจำลองทดแทนเข้าไปในกระบวนการคำนวณหลักของขั้นตอนวิธีเหล่านั้น

กระบวนการผสมผสานแบบจำลองทดแทนในขั้นตอนวิธีพัฒนาการสามารถทำได้หลายรูปแบบ โดยการใช้ขั้นตอนวิธีพัฒนาการนั้นๆ เป็นกระบวนการคำนวณหลัก เมื่อได้ผลเฉลยจำนวนมากพอที่จะสร้างแบบจำลองทดแทนจากการดำเนินการทางวิวัฒนาการ ทำการสร้างแบบจำลองทดแทนและใช้กระบวนการหาค่าเหมาะที่สุดภายใต้แบบจำลองทดแทนนี้สร้างประชากรลูกกลุ่มพิเศษป้อนให้กับ กระบวนการคำนวณของขั้นตอนวิธีพัฒนาการหลัก การดำเนินการในลักษณะนี้ทำให้สมรรถนะในการหาผลเฉลยของขั้นตอนวิธีพัฒนาการแบบผสมมีสมรรถนะในการหาค่าตอบเพิ่มมากขึ้นอย่างมีนัยสำคัญ

ได้มีการศึกษาการนำผลเฉลยประชากรพ่อ-แม่ และประชากรลูกมาคำนวณหาเวกเตอร์เกรเดียนต์โดยประมาณของตัวแปรออกแบบ และใช้เวกเตอร์เกรเดียนต์นั้นเป็นเวกเตอร์กำหนดทิศทางในการหาผลเฉลย นอกจากนี้ ได้พัฒนาตัวดำเนินการที่ใช้ทิศทางวิวัฒนาการเพื่อใช้งานร่วมกับ เวกเตอร์เกรเดียนต์โดยประมาณ ผลการทดสอบสมรรถนะแสดงให้เห็นว่า เกรเดียนต์ประมาณที่คำนวณจากสมาชิกของประชากรในขั้นตอนวิธีพัฒนาการ มีประสิทธิภาพในการหาผลเฉลยเหมาะที่สุดสูงมาก

ในส่วนของการประยุกต์ใช้งานขั้นตอนวิธีแบบผสมกับการออกแบบระบบจริงทางวิศวกรรม ได้ทำการประยุกต์ใช้ SPEA กับแบบจำลองทดแทนหลายรูปแบบเพื่อออกแบบชิ้นส่วนทางกล ในที่นี้คือแขนรับโมเมนต์บิดที่ทำจากโลหะแผ่น ผลจากการออกแบบและทดสอบสมรรถนะของแบบจำลองพบว่า

แบบจำลองฟังก์ชันกำลังสองและแบบจำลอง Kriging ให้ผลเฉลยพารามิเตอร์ที่ดีที่สุด ทั้งนี้ เป็นเพราะว่าแบบจำลองทั้งสองทำนายค่าฟังก์ชันการออกแบบแม่นยำที่สุด ในการออกแบบชิ้นส่วนยานยนต์และการออกแบบแขนรับโมเมนต์บิดสามมิติพบว่า การประยุกต์ใช้ PBIL ที่มีแบบจำลองทดแทนให้ผลการออกแบบดีกว่าการใช้ PBIL แบบไม่มีแบบจำลองทดแทนโดยที่ใช้จำนวนครั้งในการวิเคราะห์ทางไฟไนต์เอลิเมนต์น้อยกว่าอย่างมาก

การออกแบบรูปร่างของอุปกรณ์ระบายความร้อนเป็นกระบวนการออกแบบที่ทำได้ยาก ส่วนมากจะได้คำตอบจากการกำหนดรูปร่างของอุปกรณ์มาก่อน แล้วทำการหาขนาดที่เหมาะสมที่สุดเท่านั้น ในงานนี้ได้นำเสนอกระบวนการถอดรหัสตัวแปรออกแบบสำหรับอุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์ตรง อุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์เฉียง อุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์ตรง และอุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์เฉียง กระบวนการออกแบบอุปกรณ์ทั้งสี่นี้ จัดว่าเป็นทิศทางใหม่ในการออกแบบอุปกรณ์ระบายความร้อน ผลการออกแบบพบว่า อุปกรณ์ระบายความร้อนแบบที่สี่ มีสมรรถนะทางความร้อนและลดค่าดำเนินการได้มากที่สุด รองลงมาคืออุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์ตรง และกระบวนการที่ได้อันดับที่สามคือ อุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์เฉียง ผลการผสมผสานแบบจำลองทดแทนกับขั้นตอนวิธีวิวัฒนาการเพื่อการออกแบบอุปกรณ์นี้พบว่า การผสมผสานแบบจำลองทดแทนช่วยให้สมรรถนะของขั้นตอนวิธีวิวัฒนาการเหล่านั้นมีประสิทธิภาพสูงขึ้นและยังใช้จำนวนครั้งในการวิเคราะห์ไฟไนต์เอลิเมนต์น้อยกว่าการรันขั้นตอนวิธีวิวัฒนาการที่ไม่มีแบบจำลองทดแทนอย่างมาก

คำหลัก: ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย, แบบจำลองทดแทน, การออกแบบชิ้นส่วนทางกล, การออกแบบอุปกรณ์ระบายความร้อน, เกรเดียนต์โดยประมาณ, ขั้นตอนวิธีผสมผสาน

Abstract

Project Code: RSA5280006

Project Title: Performance Enhancement of Multiobjective Evolutionary Algorithms for Design Problems with Expensive Function Evaluation

**Investigator: Sujin Bureerat, Department of Mechanical Engineering
Faculty of Engineering, Khon Kaen University**

Email: sujbur@kku.ac.th

Project Period 3 years

This project investigates performance enhancement of multiobjective evolutionary algorithms (MOEAs) dealing with design problems with expensive function evaluation. Firstly, such performance improvement of MOEAs is achieved in such a way that surrogate models are integrated into the main procedure of MOEAs leading to hybrid algorithms. The obtained hybrid algorithms are then implemented on real-world applications that have time-consuming function evaluation. Hybridisation of MOEAs with surrogate models can be carried out by using the main search procedure of a particular MOEA is used as the main optimisation procedure. Having produced a number of design solutions, a surrogate model can be constructed. Then a MOEA is used to search for Pareto optimum solutions while function evaluation is based on the surrogate model. This means that the surrogate-based optimisation is a special operator to produce some offspring to the main optimisation process.

Approximation of function gradients from those already explored design solutions during an evolutionary optimisation process is proposed in this study. It has been found that the integration of approximate gradients to the evolutionary procedure leads to an improved EA performance. This newly invented operator will be further investigated in the future research.

Design of mechanical parts using MOEAs has been demonstrated. Those components include a torque arm plate, an automotive connecting part, and a three-dimensional torque arm. Function evaluation for designing the components is accomplished by means of finite element analysis which is said to be time consuming. It has been found that, with the interfacing of surrogate models into MOEA search procedure, the performance of the optimisation process has been greatly improved in terms of both convergence rate and computing time.

Design/optimisation of air-cooled heatsinks using MOEAs has been demonstrated. This is said to be a difficult and complicated design problem that is impossible to be tackled by using classical gradient-based optimisers. The encoding/decoding processes for heatsinks with straight plate fins, oblique plate fins, straight pin fins, and splayed pin fins are proposed. The design problem is to simultaneously minimise heatsink junction temperature and fan pumping power. A number of MOEAs are used to tackle the design problems and it is found that population-based incremental learning is the most efficient MOEA for this design task. It is also illustrated that the integration of surrogate models into the MOEA main procedure leads to a superior heatsink design process.

Keywords: Multiobjective Evolutionary Algorithms, Surrogate Models, Mechanical Part Design, Heatsink Design, Approximate Gradient, Hybrid Algorithms

หน้าสรุปโครงการ (Executive Summary)

1. ชื่อโครงการ (ภาษาไทย) การเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายสำหรับปัญหาการออกแบบที่ใช้เวลามากในการหาค่าฟังก์ชัน

(ภาษาอังกฤษ) Performance Enhancement of Multiobjective Evolutionary Algorithms for Design Problems with Expensive Function Evaluation

2. ชื่อหัวหน้าโครงการ หน่วยงานที่สังกัด ที่อยู่ หมายเลขโทรศัพท์ โทรสาร และ e-mail

หัวหน้าโครงการ (ชื่อ-สกุล ภาษาไทย) สุจินต์ บุรีรัตน์

(ชื่อ-สกุล ภาษาอังกฤษ) Sujin Bureerat

สถานที่ทำงาน

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

จังหวัด ขอนแก่น รหัสไปรษณีย์ 40002

โทรศัพท์ 043-202845 โทรสาร 043-202849

โทรศัพท์มือถือ 09-7114229 e-mail sujbur@kku.ac.th

3. สาขาวิชาที่ทำการวิจัย

Multiobjective Optimisation

5. ระยะเวลาดำเนินงาน 3 ปี

6. ได้เสนอโครงการนี้ หรือโครงการที่มีส่วนเหมือนกับเรื่องนี้บางส่วนเพื่อขอทุนต่อแหล่งทุนอื่นที่ใดบ้าง
✓ ไม่ได้เสนอต่อแหล่งทุนอื่น

7. ปัญหาที่ทำการวิจัย และความสำคัญของปัญหา

ปัญหาการหาค่าเหมาะที่สุด (optimisation problem) คือปัญหาทางคณิตศาสตร์ที่ต้องการหาผลเฉลยการออกแบบหรือเวกเตอร์ของตัวแปรออกแบบ (design variables) ที่ให้ค่าฟังก์ชันเป้าหมาย (objective function) เหมาะที่สุด (ต่ำสุดหรือสูงสุด) และต้องสอดคล้องกับเงื่อนไขบังคับ (constraints) การหาผลเฉลยของปัญหาการหาค่าเหมาะที่สุดเป้าหมายเดียวจะได้ผลเฉลยหนึ่งคำตอบแต่ผลเฉลยของปัญหาการออกแบบหลายเป้าหมายจะมีจำนวนเป็นอนันต์เรียกว่าเซตคำตอบของพาเรโต (Pareto Optimum Set) ซึ่งการตัดสินใจเลือกคำตอบใดๆ ในเซตคำตอบดังกล่าวเพื่อการใช้งานจริงขึ้นอยู่กับผู้ออกแบบหรือบุคลากรที่เกี่ยวข้อง

ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย (multiobjective evolutionary algorithms, MOEAs) คือวิธีหาผลเฉลยเหมาะที่สุดของพาเรโตของปัญหาการหาค่าเหมาะที่สุดแบบหลายเป้าหมาย โดยใช้หลักการ

เลียนแบบวิวัฒนาการของระบบบางอย่างในธรรมชาติและการสุ่ม (randomization) ผสมกับแผนการการคัดสรรผลเฉลยที่ไม่ถูกรอบงำ (non-dominated sorting scheme) ข้อดีของการประยุกต์ใช้วิธีหาค่าเหมาะที่สุดเหล่านี้คือ มีความง่ายในการใช้งาน มีกระบวนการหาค่าเหมาะที่สุดที่คงทน (robust) เป็นวิธีที่สามารถใช้หาผลเฉลยของปัญหาการออกแบบเกือบทุกรูปแบบ ทั้งนี้เป็นเพราะว่าวิธีวิวัฒนาการไม่ได้ใช้อนุพันธ์ของฟังก์ชันในการหาผลเฉลย และจุดเด่นที่สุดที่ทำให้วิธีวิวัฒนาการเป็นที่นิยมมากคือความสามารถในหาเซตของผลเฉลยของพารามิเตอร์ได้จากการดำเนินการหาค่าเหมาะที่สุด (optimization run) เพียงครั้งเดียว ในขณะที่วิธีหาค่าเหมาะที่สุดอื่นจะต้องดำเนินการเป็นจำนวนหลายครั้งเท่ากับจำนวนของผลเฉลยเหมาะที่สุดของพารามิเตอร์ที่ต้องการ อย่างไรก็ตาม วิธีวิวัฒนาการแบบหลายเป้าหมายยังมีข้อเสียที่สำคัญมากกล่าวคือ มีอัตราการลู่เข้า (convergence rate) ต่ำ และไม่มีความสม่ำเสมอ (consistency) ในการลู่เข้า (ไม่สามารถหาชุดผลเฉลยเดิมได้ในการรันหลายรอบ) และไม่สามารถรับประกันได้ว่าผลเฉลยที่ได้เป็นผลเฉลยเหมาะที่สุดจริงหรือไม่

ในการประยุกต์ใช้การหาค่าเหมาะที่สุดสำหรับการออกแบบระบบทางวิศวกรรม ผู้ออกแบบมักเจอปัญหาการออกแบบที่มีฟังก์ชันเป้าหมายมากกว่าหนึ่งเป้าหมาย ตัวอย่างเช่น ในปัญหาการออกแบบชิ้นส่วนยานยนต์หรือโครงสร้างทางกลอื่นๆ ที่มีรูปทรงสลับซับซ้อน ฟังก์ชันเป้าหมายโดยทั่วไปคือน้ำหนักของโครงสร้าง อย่างไรก็ตาม เมื่อลดน้ำหนักของโครงสร้างซึ่งเป็นการลดขนาดของชิ้นส่วนย่อมส่งผลให้ความแข็งแรงของโครงสร้างลดลง ดังนั้น ในบางกรณี การออกแบบโครงสร้างจึงมีฟังก์ชันเป้าหมายเป็นการหาความแข็งแรงของโครงสร้างสูงสุด ซึ่งค่าความแข็งแรงของโครงสร้างนั้นสามารถแบ่งได้เป็นความแข็งแรงเชิงสถิต (static stiffness) และความแข็งแรงเชิงพลวัต (dynamic stiffness) ซึ่งสามารถแสดงในรูปของค่าความถี่ธรรมชาติของชิ้นงาน ในปัญหาการออกแบบชิ้นส่วนระบายความร้อน (heat sink) ของชิปของคอมพิวเตอร์ ฟังก์ชันเป้าหมายที่นักออกแบบต้องการหาค่าเหมาะที่สุดคือสมรรถนะเชิงความร้อน (thermal performance) และราคาในการดำเนินการระบายความร้อน (operating cost)

การหาค่าฟังก์ชันของระบบทางวิศวกรรมเหล่านี้จำเป็นต้องใช้วิธีคำนวณเชิงตัวเลขขั้นสูงเช่น วิธีไฟไนต์เอลิเมนต์ (finite element methods) หรือวิธีไฟไนต์วอลุ่ม (finite volume methods) ซึ่งเป็นการวิเคราะห์ทั้งในแบบที่เป็นและไม่เป็นเชิงเส้น (linear and non-linear) ด้วยการวิเคราะห์โดยใช้วิธีคำนวณเชิงตัวเลขเหล่านี้ ทำให้ฟังก์ชันออกแบบมีความซับซ้อนส่งผลให้เกิดความยากหรือแม้กระทั่งเป็นไปได้เลยในการคำนวณหาอนุพันธ์ของฟังก์ชัน ด้วยเหตุนี้ การประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายในการหาผลเฉลยของพารามิเตอร์จึงเป็นทางเลือกที่เหมาะสมกับปัญหาที่เกิดขึ้น อย่างไรก็ตาม การวิเคราะห์เชิงตัวเลขเหล่านี้เมื่อประยุกต์ใช้กับปัญหาการออกแบบจริงซึ่งมีความสลับซับซ้อนจะใช้เวลาในการคำนวณนานมาก ทั้งนี้เป็นเพราะว่ากริดที่ใช้ต้องมีความละเอียดสูง ส่งผลให้ระดับความเป็นอิสระ (degree of freedom) ของปัญหามีค่าสูง และบางกรณีเป็นการวิเคราะห์ที่มีความไม่เป็นเชิงเส้นสูง การคำนวณที่ใช้เวลานานนี้เป็นอุปสรรคต่อการประยุกต์ใช้วิธีวิวัฒนาการซึ่งโดยทั่วไปมีอัตราการลู่เข้าต่ำส่งผลให้การดำเนินการหาค่าเหมาะที่สุดต้องใช้เวลาหน่วยเป็น วัน สัปดาห์ หรือแม้กระทั่งเดือน ซึ่งเป็นสิ่งที่ผู้ออกแบบไม่พึงประสงค์

วัตถุประสงค์ของโครงการวิจัยนี้คือการพัฒนาและเพิ่มสมรรถนะในการหาผลเฉลยของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย เพื่อสามารถใช้งานจริงทางวิศวกรรมเครื่องกลได้ รายละเอียดของงานจะเป็นการผสมผสาน (hybridization) ระหว่างวิธีวิวัฒนาการแบบหลายเป้าหมายที่พัฒนาขึ้นใหม่หรือวิธีเดิมที่มีอยู่แล้วกับแนวคิดของวิธีหาค่าเหมาะที่สุดที่ใช้อนุพันธ์ (derivative-based optimizers, DBO) และแบบจำลองพื้นผิวตอบสนอง (response surface models, RSM) นอกจากนี้ จะมีการพัฒนาการประมาณค่าเวกเตอร์เกรเดียนต์ (Approximate gradient, APG) จากกลุ่มประชากรของวิธีวิวัฒนาการเพื่อเป็นเวกเตอร์ทิศทางในการนำประชากรไปสู่ผลเฉลยเหมาะที่สุด และจะมีการทดสอบการนำข้อดีของวิธี simplex downhill เพื่อเพิ่มสมรรถนะในการหาผลเฉลยของวิธีวิวัฒนาการ

กระบวนการออกแบบที่พัฒนาขึ้นมาใหม่จะถูกประยุกต์ใช้งานกับการออกแบบหาค่าเหมาะที่สุดของระบบทางกลที่กล่าวถึงในข้างต้น ซึ่งประกอบด้วย การออกแบบชิ้นส่วนระบายความร้อนของซีพียู คอมพิวเตอร์ และการออกแบบชิ้นส่วนทางกลและชิ้นส่วนยานยนต์ ผลที่คาดว่าจะได้รับจากโครงการนี้คือกระบวนการหาค่าเหมาะที่สุดแบบหลายเป้าหมายที่มีสมรรถนะในการหาค่าตอบสูงกว่ากระบวนการที่มีมาก่อน ได้กระบวนการออกแบบชิ้นส่วนยานยนต์ และกระบวนการออกแบบชิ้นส่วนระบายความร้อนของซีพียูคอมพิวเตอร์ที่มีประสิทธิภาพสูง ผลจากการวิจัยนี้จะช่วยเพิ่มความก้าวหน้าให้กับเทคโนโลยีการออกแบบเหมาะที่สุดแบบมีหลายฟังก์ชันเป้าหมาย การออกแบบอุปกรณ์แลกเปลี่ยนความร้อน (heat exchangers)

8. วัตถุประสงค์

8.1 เพื่อพัฒนากระบวนการหาค่าเหมาะที่สุดแบบหลายเป้าหมายที่มีสมรรถนะในการหาค่าตอบสูง โดยการเชื่อมโยงระหว่าง ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย แบบจำลองพื้นผิวตอบสนอง และวิธีอื่นๆ

8.2 เพื่อศึกษาแนวทางในการเพิ่มประสิทธิภาพของกระบวนการออกแบบชิ้นส่วนยานยนต์และชิ้นส่วนทางกล

8.3 เพื่อศึกษาแนวทางการพัฒนากระบวนการออกแบบชิ้นส่วนระบายความร้อนของซีพียูคอมพิวเตอร์โดยการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการ

9. ระเบียบวิธีวิจัย

9.1 พัฒนากระบวนการหาค่าเหมาะที่สุดแบบหลายเป้าหมายที่เกิดจากการผสมผสานระหว่าง ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย แบบจำลองพื้นผิวตอบสนอง และแนวคิดของวิธีหาค่าเหมาะที่สุดแบบใช้อนุพันธ์ การดำเนินการในขั้นตอนนี้จะเป็นการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายเพื่อเป็นตัวดำเนินการหลักของกระบวนการออกแบบหาค่าเหมาะที่สุด ใช้แบบจำลองพื้นผิวตอบสนองเพื่อประมาณค่าฟังก์ชันเป้าหมายและเงื่อนไขบังคับ และทำการประยุกต์ใช้วิธีหาค่าเหมาะที่สุดแบบใช้อนุพันธ์ในการเพิ่มสมรรถนะการหาผลเฉลยของปัญหาการออกแบบที่ใช้พื้นผิวตอบสนองในการคำนวณค่าของ

สารบัญ

กิตติกรรมประกาศ	3
บทคัดย่อ	4
Abstract	6
หน้าสรุปโครงการ	7
บทที่ 1 บทนำ	12
บทที่ 2 การหาค่าเหมาะที่สุดหลายเป้าหมายด้วยขั้นตอนวิธีวิวัฒนาการ	18
บทที่ 3 การเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการด้วยแบบจำลองทดแทน	32
บทที่ 4 การเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการด้วยค่าประมาณของเกรเดียนต์ และวิธีหิมเพิลล์ลงเขา	38
บทที่ 5 การออกแบบชิ้นส่วนทางกลและชิ้นส่วนยานยนต์	52
บทที่ 6 การออกแบบอุปกรณ์ระบายความร้อนแบบครีบก้น	75
บทที่ 7 การออกแบบอุปกรณ์ระบายความร้อนแบบครีบก้นตรงโดยใช้ SPEA ร่วมกับ แบบจำลองพื้นผิวตอบสนอง	92
บทที่ 8 การออกแบบหาค่าเหมาะที่สุดแบบหลายเป้าหมายของอุปกรณ์ระบายความร้อน แบบครีบก้นแผ่นเอียง	106
บทที่ 9 การออกแบบอุปกรณ์ระบายความร้อนแบบครีบก้น	119
บทที่ 10 สรุป วิจัย และงานต่อเนื่อง	144
เอกสารอ้างอิง	146
Output จากโครงการวิจัย	151
ภาคผนวก	153

บทนำ

1.1 ความสำคัญและที่มาของการวิจัย

ปัญหาการหาค่าเหมาะที่สุด (optimization problem) คือปัญหาทางคณิตศาสตร์ที่ต้องการหาผลเฉลยการออกแบบหรือเวกเตอร์ของตัวแปรออกแบบ (design variables) ที่ให้ค่าฟังก์ชันเป้าหมาย (objective function) เหมาะที่สุด (ต่ำสุดหรือสูงสุด) และต้องสอดคล้องกับเงื่อนไขบังคับ (constraints) การหาผลเฉลยของปัญหาการหาค่าเหมาะที่สุดเป้าหมายเดียวจะได้ผลเฉลยหนึ่งคำตอบ แต่ผลเฉลยของปัญหาออกแบบหลายเป้าหมายจะมีจำนวนเป็นอนันต์ เรียกว่าเซตคำตอบของพาเรโต (Pareto Optimum Set) ซึ่งการตัดสินใจเลือกคำตอบใดๆ ในเซตคำตอบดังกล่าวเพื่อการใช้งานจริงขึ้นอยู่กับผู้ออกแบบหรือบุคลากรที่เกี่ยวข้อง

ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย (multiobjective evolutionary algorithms, MOEAs) คือวิธีหาผลเฉลยเหมาะที่สุดของพาเรโตของปัญหาการหาค่าเหมาะที่สุดแบบหลายเป้าหมาย โดยใช้หลักการเลียนแบบวิวัฒนาการของระบบบางอย่างในธรรมชาติและการสุ่ม (randomization) ผสมกับการคัดสรรผลเฉลยที่ไม่ถูกรอบงำ (non-dominated sorting scheme) ข้อดีของการประยุกต์ใช้วิธีหาค่าเหมาะที่สุดเหล่านี้คือ มีความง่ายในการใช้งาน มีกระบวนการหาค่าเหมาะที่สุดที่คงทน (robust) เป็นวิธีที่สามารถใช้หาผลเฉลยของปัญหาการออกแบบเกือบทุกรูปแบบ ทั้งนี้เป็นเพราะว่าวิธีวิวัฒนาการไม่ได้ใช้อนุพันธ์ของฟังก์ชันในการหาผลเฉลย และจุดเด่นที่สุดที่ทำให้วิธีวิวัฒนาการเป็นที่นิยมมากคือความสามารถในหาเซตของผลเฉลยของพาเรโตได้จากการดำเนินการหาค่าเหมาะที่สุด (optimization run) เพียงครั้งเดียว ในขณะที่วิธีหาค่าเหมาะที่สุดอื่นจะต้องดำเนินการเป็นจำนวนหลายครั้งเท่ากับจำนวนของผลเฉลยเหมาะที่สุดของพาเรโตที่ต้องการ อย่างไรก็ตาม วิธีวิวัฒนาการแบบหลายเป้าหมายยังมีข้อเสียที่สำคัญมากกล่าวคือ มีอัตราการลู่เข้า (convergence rate) ต่ำ และไม่มีความสม่ำเสมอ (consistency) ในการลู่เข้า (ไม่สามารถหาชุดผลเฉลยเดิมได้ในการรันหลายรอบ) และไม่สามารถรับประกันได้ว่าผลเฉลยที่ได้เป็นผลเฉลยเหมาะที่สุดจริงหรือไม่

ในการประยุกต์ใช้การหาค่าเหมาะที่สุดสำหรับการออกแบบระบบทางวิศวกรรม ผู้ออกแบบมักเจอปัญหาการออกแบบที่มีฟังก์ชันเป้าหมายมากกว่าหนึ่งเป้าหมาย ตัวอย่างเช่น ในปัญหาการออกแบบชิ้นส่วนยานยนต์หรือโครงสร้างทางกลอื่นๆ ที่มีรูปทรงสลับซับซ้อน ฟังก์ชันเป้าหมายโดยทั่วไปคือน้ำหนักของโครงสร้าง อย่างไรก็ตาม เมื่อลดน้ำหนักของโครงสร้างซึ่งเป็นการลดขนาดของชิ้นส่วนย่อมส่งผลให้ความแข็งแรงของโครงสร้างลดลง ดังนั้น ในบางกรณี การออกแบบโครงสร้างจึงมีฟังก์ชันเป้าหมายเป็นการหาความแข็งแรงของโครงสร้างสูงสุด ซึ่งค่าความแข็งแรงของโครงสร้างนั้นสามารถแบ่งได้เป็นความแข็งแรงเชิงสถิต (static stiffness) และความแข็งแรงเชิงพลวัต (dynamic stiffness) ซึ่งสามารถแสดงในรูปของ

ค่าความถี่ธรรมชาติของชิ้นงาน ในปัญหาการออกแบบชิ้นส่วนระบายความร้อน (heat sink) ของซีพียูของคอมพิวเตอร์ ฟังก์ชันเป้าหมายที่นักออกแบบต้องการหาค่าเหมาะที่สุดคือสมรรถนะเชิงความร้อน (thermal performance) และราคาในการดำเนินการระบายความร้อน (operating cost)

การหาค่าฟังก์ชันของระบบทางวิศวกรรมเหล่านี้จำเป็นต้องใช้วิธีคำนวณเชิงตัวเลขขั้นสูงเช่น วิธีไฟไนต์เอลิเมนต์ (finite element methods) หรือวิธีไฟไนต์โวลูม (finite volume methods) ซึ่งเป็นการวิเคราะห์ทั้งในแบบที่เป็นและไม่เป็นเชิงเส้น (linear and non-linear) ด้วยการวิเคราะห์โดยใช้วิธีคำนวณเชิงตัวเลขเหล่านี้ ทำให้ฟังก์ชันออกแบบมีความซับซ้อนส่งผลให้เกิดความยากหรือแม้กระทั่งเป็นไปได้เลยในการคำนวณหาอนุพันธ์ของฟังก์ชัน ด้วยเหตุนี้ การประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายในการหาผลเฉลยของพารามิเตอร์ที่เป็นทางออกที่เหมาะสมกับปัญหาที่เกิดขึ้น อย่างไรก็ตาม การวิเคราะห์เชิงตัวเลขเหล่านี้เมื่อประยุกต์ใช้กับปัญหาการออกแบบจริงซึ่งมีความสลับซับซ้อนจะใช้เวลาในการคำนวณนานมาก ทั้งนี้เป็นเพราะว่ากริดที่ใช้ต้องมีความละเอียดสูง ส่งผลให้ระดับความเป็นอิสระ (degree of freedom) ของปัญหามีค่าสูง และบางกรณีเป็นการวิเคราะห์ที่ไม่เป็นเชิงเส้นสูง การคำนวณที่ใช้เวลานานนี้เป็นอุปสรรคต่อการประยุกต์ใช้วิธีวิวัฒนาการซึ่งโดยทั่วไปมีอัตราการลู่เข้าต่ำส่งผลให้การดำเนินการหาค่าเหมาะที่สุดต้องใช้เวลาหน่วยเป็น วัน สัปดาห์ หรือแม้กระทั่งเดือน ซึ่งเป็นสิ่งที่ผู้ออกแบบไม่พึงประสงค์

วัตถุประสงค์ของโครงการวิจัยนี้คือการพัฒนาและเพิ่มสมรรถนะในการหาผลเฉลยของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย เพื่อสามารถใช้งานจริงทางวิศวกรรมเครื่องกลได้ รายละเอียดของงานจะเป็นการผสมผสาน (hybridization) ระหว่างวิธีวิวัฒนาการแบบหลายเป้าหมายที่พัฒนาขึ้นมาใหม่หรือวิธีเดิมที่มีอยู่แล้วกับแนวคิดของวิธีหาค่าเหมาะที่สุดที่ใช้อนุพันธ์ (derivative-based optimizers, DBO) และแบบจำลองพื้นผิวตอบสนอง (response surface models, RSM) นอกจากนี้ จะมีการพัฒนาการประมาณค่าเวกเตอร์เกรเดียนต์ (Approximate gradient, APG) จากกลุ่มประชากรของวิธีวิวัฒนาการเพื่อเป็นเวกเตอร์ทิศทางในการนำประชากรไปสู่ผลเฉลยเหมาะที่สุด และจะมีการทดสอบการนำข้อดีของวิธี simplex downhill เพื่อเพิ่มสมรรถนะในการหาผลเฉลยของวิธีวิวัฒนาการ

กระบวนการออกแบบที่พัฒนาขึ้นมาใหม่จะถูกประยุกต์ใช้งานกับการออกแบบหาค่าเหมาะที่สุดของระบบทางกลที่กล่าวถึงในข้างต้น ซึ่งประกอบด้วย การออกแบบชิ้นส่วนระบายความร้อนของซีพียูคอมพิวเตอร์ และการออกแบบชิ้นส่วนทางกลและชิ้นส่วนยานยนต์ ผลที่คาดว่าจะได้รับจากโครงการนี้คือกระบวนการหาค่าเหมาะที่สุดแบบหลายเป้าหมายที่มีสมรรถนะในการหาค่าตอบสูงกว่ากระบวนการที่มีมาก่อน ได้กระบวนการออกแบบชิ้นส่วนยานยนต์ และกระบวนการออกแบบชิ้นส่วนระบายความร้อนของซีพียูคอมพิวเตอร์ที่มีประสิทธิภาพสูง ผลจากการวิจัยนี้จะช่วยเพิ่มความก้าวหน้าให้กับเทคโนโลยีการออกแบบเหมาะที่สุดแบบมีหลายฟังก์ชันเป้าหมาย การออกแบบอุปกรณ์แลกเปลี่ยนความร้อน (heat exchangers)

1.2 ผลงานวิจัยที่เกี่ยวข้อง

ขั้นตอนวิธีวิวัฒนาการถูกพัฒนาขึ้นมาสำหรับการหาค่าเหมาะที่สุดมานานกว่า 50 ปี วิธีวิวัฒนาการในยุคเริ่มต้นเป็นวิธีหาค่าเหมาะที่สุดสำหรับปัญหาการออกแบบที่มีฟังก์ชันเป้าหมายเดียว เช่น genetic algorithms (GA), evolutionary programming (EP) และ evolution strategies (ES) ต่อมาได้มีการพัฒนาเทคโนโลยีทางด้านนี้อย่างแพร่หลายทั้งทางด้านการพัฒนาวิธีวิวัฒนาการใหม่ เช่น population-based incremental learning (PBIL) และ particle swarm optimization (PSO) นอกจากนี้ยังการปรับปรุงสมรรถนะในการหาผลเฉลยของวิธีวิวัฒนาการเดิม การประยุกต์ตัวแปรออกแบบที่เป็นเลขฐานสองจำนวนเต็ม และจำนวนจริง เช่น stud genetic algorithm (Stud-GA) และ breeder genetic algorithm (BGA) เป็นต้น และได้มีการประยุกต์ใช้ในงานออกแบบจริงในหลากหลายสาขาวิชา เช่น ในปัจจุบันขั้นตอนวิธีวิวัฒนาการได้มีการพัฒนาแบบก้าวกระโดด ทำให้คำว่าขั้นตอนวิธีวิวัฒนาการไม่ครอบคลุมเนื้อหาทั้งหมด ดังนั้นจึงมีคำศัพท์เกิดขึ้นกล่าวคือวิธีศึกษาสำนึกขั้นสูง (meta-heuristics, MHs) อย่างไรก็ตาม ในรายงานวิจัยนี้จะยังคงใช้คำว่าขั้นตอนวิธีวิวัฒนาการเพราะเป็นคำที่ใช้มาตั้งแต่เริ่มต้น วิธีศึกษาสำนึกขั้นสูงหรือขั้นตอนวิธีวิวัฒนาการทั้งแบบเป้าหมายเดียวหรือแบบหลายเป้าหมายที่พัฒนาร่วมสมัย (state-of-the-art) มีดังตัวอย่างต่อไปนี้ unrestricted population size evolutionary optimization algorithm (UPSEMOA), multiobjective harmony search, multiobjective bee colony optimization, charged system search, real-code ant colony optimization, league championship algorithm, cuckoo search, teaching-learning-based method และ real-code population-based incremental learning เป็นต้น

การพัฒนาขั้นตอนวิธีวิวัฒนาการเพื่อหาผลเฉลยของพารามิเตอร์ของปัญหาการหาค่าเหมาะที่สุดแบบหลายเป้าหมายได้เริ่มขึ้นเมื่อประมาณ 20 ปีที่ผ่านมา ในยุคแรก วิธีวิวัฒนาการแบบหลายเป้าหมายถูกพัฒนาเพียงเพื่อการใช้งานได้กล่าวคือ สามารถหาเซตผลเฉลยของพารามิเตอร์ได้ภายในการดำเนินการหาค่าเหมาะที่สุดเพียงครั้งเดียว (1 optimization run) ตัวอย่างของวิธีดังกล่าวคือ vector evaluation genetic Algorithm (VEGA), multiobjective genetic algorithm (MOGA), strength Pareto evolutionary algorithm (SPEA) และ none-dominated sorting genetic algorithm (NSGA) ยุคที่สองของการพัฒนานั้นไปที่การเพิ่มสมรรถนะในการหาผลเฉลยของวิธีเหล่านั้น ทำให้เกิดวิธีวิวัฒนาการแบบหลายเป้าหมายชนิดใหม่ขึ้นมามากมายและยังมีการปรับปรุงวิธีการเดิมให้มีสมรรถนะสูงขึ้น ตัวอย่างของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายในยุคที่สองคือ strength Pareto evolutionary algorithm version 2 (SPEA2), none-dominated sorting genetic algorithm version 2 (NSGAII), multiobjective population-based incremental learning (PBIL), multiobjective particle swarm optimization (MPSO) และ Pareto archive evolution strategy (PAES) การพัฒนาขั้นตอนวิธีวิวัฒนาการแบบใหม่นั้นเรียกได้ว่าเกิดขึ้นทุกวันในทุกมุมโลก ควบคู่ไปกับการพัฒนาเพื่อเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย จึงได้มีการพัฒนากระบวนการในการเปรียบเทียบสมรรถนะในการหาผลเฉลยของวิธีวิวัฒนาการต่างๆ เพื่อสามารถระบุวิธีที่ดีที่สุดได้

ยุคที่สามคือยุคปัจจุบัน เป็นยุคที่เน้นหนักไปที่การประยุกต์ขั้นตอนวิธีการเหล่านี้เพื่อใช้กับงาน ออกแบบจริงในสาขาวิชาต่างๆ อย่างไรก็ตาม ข้อด้อยของวิธีการเหล่านี้ที่ยังไม่สามารถแก้ไขได้อย่าง สมบูรณ์แบบคือ การมีอัตราการลู่เข้าต่ำ ไม่มีความสม่ำเสมอในการลู่เข้า และไม่สามารถรับประกันได้ว่าผล เฉลยที่ได้มาคือผลเฉลยเหมาะที่สุดจริง ทั้งนี้เป็นเพราะว่าขั้นตอนวิธีเหล่านี้ใช้การสุ่ม (randomization) ใน การค้นหาผลเฉลย ด้วยเหตุที่วิธีวิวัฒนาการแบบหลายเป้าหมายมีอัตราการลู่เข้าต่ำ ทำให้มีความยุ่งยากในการ ประยุกต์ใช้วิธีวิวัฒนาการแบบหลายเป้าหมายกับปัญหาการออกแบบที่ใช้เวลานานมากในการวิเคราะห์หาค่า ฟังก์ชัน การแก้ไขปัญหานั้นสามารถทำได้โดยการเพิ่มแบบจำลองพื้นผิวดตอบสนองเข้ามาใน กระบวนการคำนวณของวิธีวิวัฒนาการ ดังนั้นการผสมผสานระหว่างขั้นตอนวิธีวิวัฒนาการแบบหลาย เป้าหมายและแบบจำลองพื้นผิวดตอบสนองเพื่อประโยชน์หลายๆด้านของการหาค่าเหมาะที่สุดจึงเป็นหนึ่งใน ประเด็นการวิจัยที่เกี่ยวข้องกับขั้นตอนวิธีวิวัฒนาการ แบบจำลองพื้นผิวดตอบสนองที่นิยมใช้ควบคู่กับ ขั้นตอนวิธีวิวัฒนาการและวิธีหาค่าเหมาะที่สุดอื่นในกระบวนการหาค่าเหมาะที่สุดคือ

- quadratic model
- polynomial regression
- radial basis function interpolation
- artificial neural networks
- Gaussian process model
- และอื่นๆ

การผสมผสานวิธีการทั้งสองนี้ ส่วนมากมักจะอยู่ในรูปแบบที่ตัวดำเนินการหาค่าเหมาะที่สุดหลักคือ วิธีวิวัฒนาการ ส่วนแบบจำลองพื้นผิวดตอบสนองใช้ในการประมาณค่าของฟังก์ชัน นั่นหมายความว่า การหา ค่าฟังก์ชันในกระบวนการหาค่าเหมาะที่สุดจะประหยัดเวลาในการคำนวณอย่างมาก ในงานวิจัยที่ทำมา ก่อนที่เกี่ยวข้องกับการประยุกต์ใช้การผสมผสานของสองวิธีการนี้ ส่วนมากจะเป็นการใช้แบบจำลองพื้นผิวด ตอบสนองประมาณค่าฟังก์ชันเป้าหมายเท่านั้น ทั้งนี้เป็นเพราะว่า ในการใช้พื้นผิวดตอบสนองประมาณค่า ฟังก์ชันเงื่อนไขบังคับ ถ้าหากไม่มีความแม่นยำพอ การใช้พื้นผิวดตอบสนองในกระบวนการหาค่าเหมาะที่สุด ด้วยวิธีวิวัฒนาการจะไม่มีประสิทธิภาพมากนัก

การออกแบบโครงสร้าง ชิ้นส่วนทางกล และชิ้นส่วนยานยนต์เป็นสาขาที่มีการเรียนการสอนและ ค้นคว้าวิจัยมากที่สุดสาขาหนึ่ง เนื่องจากการออกแบบเพื่อให้ชิ้นงานสามารถนำไปใช้ได้จริงนั้น มีปัจจัย หลายอย่างที่คำนึงถึง ในเบื้องต้น ชิ้นส่วนต้องมีราคาถูกเพื่อเหตุผลเชิงพาณิชย์ อย่างไรก็ตามการลดราคา ของชิ้นงานย่อมส่งผลให้เกิดการลดขนาดของชิ้นงานตามไปด้วยจึงมีผลกระทบต่อความแข็งแรงของชิ้นงาน นั้น ดังนั้นในการออกแบบชิ้นส่วนทางกลหรือโครงสร้าง นอกจากการประยุกต์การหาค่าต่ำสุดแล้ว ยังต้อง คำนึงถึงการหาความแข็งแรงสูงสุดของโครงสร้างด้วย นอกจากนี้ยังมีเงื่อนไขบังคับที่สำคัญอีกมากมาย เช่น เงื่อนไขบังคับที่ป้องกันไม่ให้โครงสร้างเกิดการเสียหายเนื่องจากภาระกระทำ ซึ่งสามารถจำแนกได้เป็น ปรากฏการณ์ทางกลดังนี้ ความเค้น การล้า (fatigue) การสั่นสะเทือน การโก่งตัว และความไม่เสถียรเชิง พลวัตของโครงสร้าง เป็นต้น ได้มีการการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการสำหรับปัญหาการออกแบบ ชิ้นส่วนเหล่านี้มีเป็นระยะเวลานานพอสมควร อย่างไรก็ตาม ถึงแม้จะมีข้อดีอยู่มาก อาทิเช่น การได้รับผล

เฉลยการออกแบบที่ดีเสมอ ข้อด้อยของการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการยังมีปัญหาที่ไม่สามารถหลีกเลี่ยงได้คือ อัตราการลู่เข้าต่ำส่งผลให้วิธีการเหล่านี้ไม่เหมาะสมกับปัญหาการออกแบบชิ้นส่วนทางกลที่ต้องใช้เวลาในการคำนวณฟังก์ชันในการออกแบบสูง และปัญหาที่มีตัวแปรออกแบบจำนวนมาก

ด้วยขนาดที่เล็กลงและการใช้งานที่หนักมากขึ้นของซีพียูของคอมพิวเตอร์ในยุคปัจจุบัน ทำให้มีความต้องการระบายความร้อนออกจากชิ้นส่วนซีพียูมากขึ้นเพราะความร้อนอาจก่อให้เกิดความเค้นและการเสียรูปของตัวซีพียูได้ ดังนั้น เพื่อให้ได้ชิ้นส่วนระบายความร้อนซีพียูที่ดีที่สุด จึงมีการประยุกต์ใช้การหาค่าเหมาะที่สุดในการออกแบบ การวิเคราะห์และออกแบบเหมาะที่สุดของระบบระบายความร้อนของซีพียูคอมพิวเตอร์นั้น ได้มีการศึกษาค้นคว้ามาเป็นเวลานานพอสมควร ในยุคเริ่มต้นของการออกแบบ ผู้วิจัยมักใช้สูตรที่ได้จากผลเฉลยเชิงวิเคราะห์ (analytical solution) ในการหาค่าเหมาะที่สุด ส่งผลให้โดเมนของการออกแบบไม่มีความหลากหลายมากนัก อย่างไรก็ตาม เมื่อพัฒนาการทางด้านพลศาสตร์ของไหลเชิงคำนวณ (computational fluid dynamics) อยู่ในขั้นสูง การกำหนดโดเมนของตัวแปรออกแบบสามารถทำได้หลากหลายรูปแบบส่งผลให้เกิดการยกระดับเทคโนโลยีของการออกแบบทางด้านนี้ มีการศึกษาวิจัยเกี่ยวกับชิ้นส่วนระบายความร้อนมีทั้งแบบ plate-fin , pin-fin และอื่นๆ ได้มีการประยุกต์ใช้แบบจำลองพื้นผิวดอปสันองเพื่อช่วยในการออกแบบ การประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการ และวิธีหาค่าเหมาะที่สุดแบบใช้อนุพันธ์ในการหาผลเฉลย ปัญหาการออกแบบที่มีหลายฟังก์ชันเป้าหมายของระบบทางกลนี้มักอยู่ในรูปของการหาสมรรถนะเชิงความร้อนสูงสุดและในขณะเดียวกันต้องได้ค่าการดำเนินการระบายความร้อนต่ำสุด จากงานวิจัย [78] แสดงให้เห็นว่าการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายกับปัญหาการออกแบบนี้ เป็นไปได้อย่างมีประสิทธิภาพและประสิทธิผล และสามารถขยายผลไปยังการออกแบบอุปกรณ์แลกเปลี่ยนความร้อนชนิดอื่นได้ การเพิ่มสมรรถนะในการหาผลเฉลยของขั้นตอนวิธีวิวัฒนาการย่อมส่งผลดีต่อเทคโนโลยีการออกแบบนี้

1.3 วัตถุประสงค์ของโครงการ

วัตถุประสงค์หลักของโครงการมี 3 ข้อดังนี้

1.3.1 เพื่อพัฒนากระบวนการหาค่าเหมาะที่สุดแบบหลายเป้าหมายที่มีสมรรถนะในการหาคำตอบสูงโดยการเชื่อมโยงระหว่าง ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย แบบจำลองพื้นผิวดอปสันอง และวิธีอื่นๆ

1.3.2 เพื่อศึกษาแนวทางในการเพิ่มประสิทธิภาพของกระบวนการออกแบบชิ้นส่วนยานยนต์และชิ้นส่วนทางกล

1.3.3 เพื่อศึกษาแนวทางการพัฒนากระบวนการออกแบบชิ้นส่วนระบายความร้อนของซีพียูคอมพิวเตอร์โดยการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการ

1.4 ระเบียบวิธีวิจัย

1.4.1 พัฒนาระบบการหาค่าเหมาะที่สุดแบบหลายเป้าหมายที่เกิดจากการผสมผสานระหว่าง ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย แบบจำลองพื้นผิวดตอบสนอง และแนวคิดของวิธีหาค่าเหมาะที่สุดแบบไฮออนูพันธ์ การดำเนินการในขั้นตอนนี้จะเป็นการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย เพื่อเป็นตัวดำเนินการหลักของกระบวนการออกแบบหาค่าเหมาะที่สุด ใช้แบบจำลองพื้นผิวดตอบสนองเพื่อประมาณค่าฟังก์ชันเป้าหมายและเงื่อนไขบังคับ และทำการประยุกต์ใช้วิธีหาค่าเหมาะที่สุดแบบไฮออนูพันธ์ ในการเพิ่มสมรรถนะการหาผลเฉลยของปัญหาการออกแบบที่ใช้พื้นผิวดตอบสนองในการคำนวณค่าของฟังก์ชันออกแบบ นอกจากนี้ จะมีการทดสอบการนำกลไกการหาผลเฉลยของวิธี simplex downhill เพื่อเพิ่มสมรรถนะของวิธีวิวัฒนาการ

ในการดำเนินการทดสอบเชิงตัวเลข ผู้วิจัยสามารถเลือกวิธีวิวัฒนาการแบบหลายเป้าหมาย พร้อมทั้ง ระบุชนิดของแบบจำลองพื้นผิวดตอบสนองและวิธีหาค่าเหมาะที่สุดแบบไฮออนูพันธ์ ทำให้เกิดเป็นแผนการหาค่าเหมาะที่สุด (optimization strategies) หลากหลายแผนการ จากนั้นทำการทดสอบสมรรถนะในการหาผลเฉลยของพารามิเตอร์ของแผนการเหล่านั้น เพื่อคัดเลือกแผนการหาค่าเหมาะที่สุดที่ดีที่สุด

1.4.2 ประยุกต์ใช้ระบบการหาค่าเหมาะที่สุดที่พัฒนาขึ้นมากับปัญหาการหาค่าเหมาะที่สุดของ โครงสร้างและชิ้นส่วนทางกล ในขั้นตอนนี้จะเป็นการประยุกต์ใช้แผนการหาค่าเหมาะที่สุด หลากหลายรูปแบบที่ได้จากการวิจัยในขั้นตอนที่ 9.1 กับปัญหาการหาค่าเหมาะที่สุดของโครงสร้างซึ่งต้อง ใช้การวิเคราะห์ทางไฟไนต์เอลิเมนต์ซึ่งใช้เวลามากในการคำนวณค่าฟังก์ชันการออกแบบต่างๆ

1.4.3 ประยุกต์ใช้ระบบการหาค่าเหมาะที่สุดที่พัฒนาขึ้นมากับปัญหาการหาค่าเหมาะที่สุดของชิ้นส่วน ระบายความร้อนของซีพียูคอมพิวเตอร์ เช่นเดียวกันกับการประยุกต์ใช้งานจริงในกรณีแรก ในขั้นตอนนี้เป็น การประยุกต์ใช้แผนการหาค่าเหมาะที่สุดหลากหลายวิธีเพื่อออกแบบหาค่าเหมาะที่สุดของชิ้นส่วนระบาย ความร้อนของซีพียูทั้งแบบครีบบแผ่น (plate-fin) และแบบครีบแท่ง (pin-fin) เมื่อเปรียบเทียบผลการ ออกแบบที่ได้จากวิธีเหล่านั้น ทำการสรุปหาแผนการหาค่าเหมาะที่สุดที่เหมาะสมกับปัญหาการออกแบบ หลายเป้าหมายชนิดนี้

การหาค่าเหมาะที่สุดหลายเป้าหมายด้วยขั้นตอนวิธีวิวัฒนาการ

2.1 ความนำ

การประยุกต์การออกแบบหาค่าเหมาะที่สุดสำหรับปัญหาการออกแบบจริงในงานวิศวกรรมนั้น โดยทั่วไปจะประกอบไปด้วยฟังก์ชันเป้าหมาย (objective function) และเงื่อนไขบังคับ (constraints) ตัวอย่างเช่น การออกแบบเพื่อลดน้ำหนักหรือราคาของชิ้นงานให้น้อยที่สุดแต่มีเงื่อนไขคือโครงสร้างจะต้องไม่เสียหายจากแรงกระทำ อย่างไรก็ตามในทางปฏิบัติ ฟังก์ชันเป้าหมายไม่ได้มีเฉพาะน้ำหนักหรือราคาของชิ้นงานเท่านั้น ผู้ออกแบบอาจต้องการหาน้ำหนักน้อยสุด (weight minimisation) และในขณะเดียวกันยังต้องการให้โครงสร้างของชิ้นงานมีความแข็งแรงมากที่สุด (strength maximisation) และมีเงื่อนไขบังคับคือความปลอดภัยในการใช้งาน ปัญหาลักษณะนี้เรียกว่า การหาค่าเหมาะที่สุดแบบหลายเป้าหมาย

การหาผลเฉลยของปัญหาการหาค่าเหมาะที่สุดรูปแบบนี้มีความแตกต่างจากการหาผลเฉลยของปัญหาการออกแบบที่มีฟังก์ชันเป้าหมายเดียว ในอดีต วิธีการหาค่าเหมาะที่สุดที่พัฒนาขึ้นมาส่วนใหญ่ใช้ในการหาผลเฉลยของปัญหาการออกแบบเป้าหมายเดียว อย่างไรก็ตาม เราสามารถประยุกต์ใช้วิธีเหล่านั้นกับปัญหาแบบหลายเป้าหมาย โดยการเพิ่มกระบวนการเชิงตัวเลขพิเศษ เช่น วิธี ผลรวมแบบถ่วงน้ำหนัก (weighted sum method) ข้อเสียของกระบวนการออกแบบนี้คือ เราจะต้องรันวิธีการหาค่าเหมาะที่สุดเป็นจำนวนหลายครั้งเท่ากับจำนวนผลเฉลยพारेโตที่เราต้องการ

ในปัจจุบัน ได้มีการพัฒนาวิธีการหาค่าเหมาะที่สุดแบบวิวัฒนาการหรือขั้นตอนวิธีวิวัฒนาการ เพื่อใช้ในการหาผลเฉลยเหมาะที่สุดของปัญหาแบบหลายเป้าหมาย ข้อดีของการประยุกต์ใช้วิธีการเหล่านี้คือ เป็นวิธีที่ประยุกต์ใช้งานง่าย สามารถใช้หาผลเฉลยของปัญหาการหาค่าเหมาะที่สุดได้เกือบทุกรูปแบบ เป็นวิธีหาค่าเหมาะที่สุดที่สามารถรันได้โดยไม่ต้องมีการสะดุดหรือเกิดข้อผิดพลาดในระหว่างกระบวนการคำนวณ และที่สำคัญที่สุด ขั้นตอนวิธีวิวัฒนาการที่พัฒนาขึ้นมาในปัจจุบันสามารถหากลุ่มของผลเฉลยพारेโตได้ในการรันเพียงครั้งเดียวซึ่งความสามารถตรงนี้ได้ทำให้วิธีวิวัฒนาการเป็นที่นิยมมาก อย่างไรก็ตาม วิธีการเหล่านี้ยังมีข้อเสียที่หลีกเลี่ยงไม่ได้กล่าวคือ มีอัตราการลู่เข้าช้าและไม่มีความคงเส้นคงวา (ไม่สามารถหาค่าตอบเดิมได้เสมอ ในการรันเพื่อหาผลเฉลยของปัญหาเดิมหลายครั้ง) ทั้งนี้เป็นเพราะว่า วิธีการเหล่านี้ใช้การสุ่มเป็นหลักในกระบวนการคำนวณ

ตัวอย่างขั้นตอนวิธีวิวัฒนาการที่พัฒนาขึ้นในยุคเริ่มต้นมีดังนี้

- Vector Evaluation Genetic Algorithm (VEGA)
- MultiObjective Genetic Algorithm (MOGA)
- strength Pareto evolutionary algorithm (SPEA)
- None-dominated Sorting Genetic Algorithm (NSGA)

ในยุคเริ่มต้น ผู้พัฒนากระบวนการคำนวณจะเน้นไปที่ความสามารถในการหาผลเฉลยพาเรโตได้ภายในการรันครั้งเดียว หลังจากนั้น ยุคต่อมา ได้มีการพัฒนาขั้นตอนวิธีวิวัฒนาการโดยเน้นไปที่สมรรถนะในการหาคำตอบเป็นหลัก ได้มีการพัฒนาวิธีการใหม่ และปรับปรุงวิธีการที่มีอยู่แล้วเพื่อเพิ่มสมรรถนะของวิธีเหล่านั้นเพื่อสามารถใช้งานจริงได้ ดังเช่น

non-dominated sorting genetic algorithm (NSGAI)
 strength Pareto evolutionary algorithm (SPEA2)
 multiobjective particle swarm optimisation (MPSO)
 multiobjective Population-Based Incremental Learning (PBIL)

ปัญหาการหาค่าเหมาะที่สุดแบบหลายเป้าหมาย สามารถเขียนได้ดังนี้

Find $\mathbf{x}$ such that

$$\text{Min: } \mathbf{f} = \{f_1(\mathbf{x}), \dots, f_p(\mathbf{x})\} \quad (2.1)$$

Subject to

$$\begin{aligned} g_i(\mathbf{x}) &\leq 0 \\ h_i(\mathbf{x}) &= 0 \end{aligned}$$

เมื่อ $\mathbf{x}$ คือเวกเตอร์ของตัวแปรออกแบบ

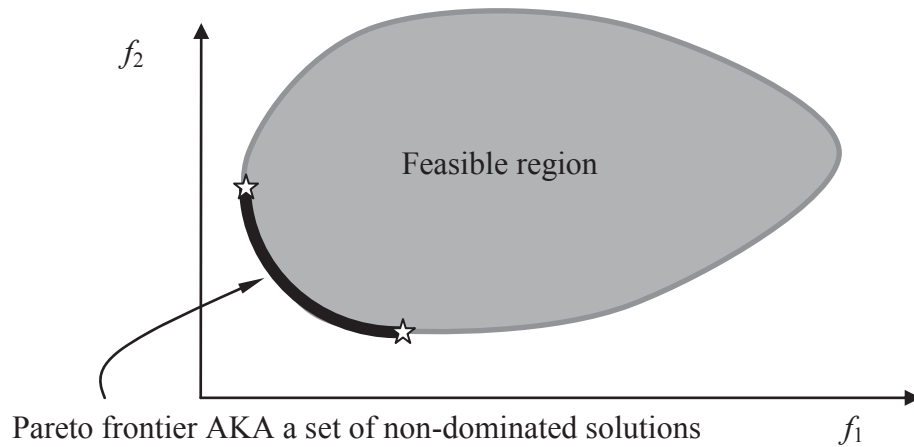
f_i คือฟังก์ชันเป้าหมาย

g_i คือเงื่อนไขบังคับแบบอสมการ

และ h_i คือเงื่อนไขบังคับแบบสมการ

โดยปกติแล้วฟังก์ชันเป้าหมายมักจะขัดแย้งกัน ตัวอย่างเช่นในกรณีของปัญหาแบบสองเป้าหมายที่ทำการหาน้ำหนักน้อยสุดและความแข็งแรงสูงสุดของโครงสร้าง ถ้าทำการเพิ่มความแข็งแรงจะส่งผลให้น้ำหนักเพิ่มขึ้น แต่ถ้าเราลดน้ำหนักจะส่งผลให้ความแข็งแรงลดลงเช่นกัน รูปที่ 2-1 แสดงพื้นที่หาคำตอบได้ (feasible region) ของปัญหาการออกแบบสองเป้าหมายบนโดเมนของฟังก์ชันเป้าหมาย จากรูปจะเห็นว่าผลเฉลยของปัญหามีมากกว่าหนึ่งผลเฉลยและเมื่อรวมผลเฉลยทั้งหมด จะได้เส้นโค้งที่แสดงถึงขอบเขตของผลเฉลยที่เหมาะสมที่สุดของพาเรโต (Pareto front) ส่วนผลเฉลย ($\mathbf{x}$) ที่อยู่บนแนวของพาเรโตเรียกว่า เซตผลเฉลยที่เหมาะสมที่สุดของพาเรโต (Pareto optimal set) ด้วยเหตุผลเช่นเดียวกับกรณีของสองฟังก์ชันเป้าหมาย แนวของพาเรโตของปัญหาแบบสามฟังก์ชันเป้าหมายจะอยู่ในรูปของพื้นผิว (surface)

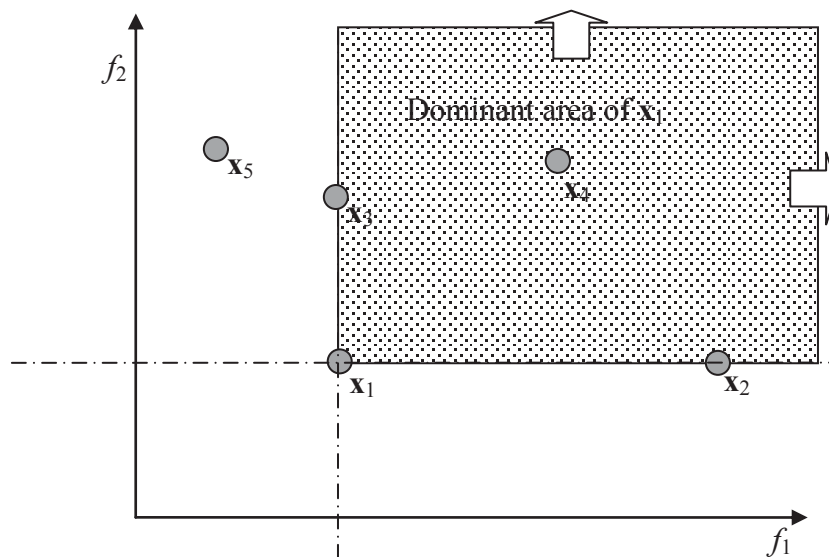
แนวคิดในการหาผลเฉลยของพาเรโตด้วยวิธีวิวัฒนาการแบบหลายเป้าหมายคือการใช้กลุ่มของผลเฉลยในการหาคำตอบ (หรือประชากร) ในแต่ละวงวน (loop) ของการค้นหาคำตอบ ในระหว่างกระบวนการค้นหาคำตอบวิธีวิวัฒนาการจะทำการเก็บผลเฉลยที่ไม่ถูกรอบงำ (non-dominated solutions) ไว้ในเมตริกซ์เก็บผลเฉลยพาเรโต (Pareto archive) นิยามสำคัญสำหรับการค้นหาคำตอบด้วยหลักการคัดสรรผลเฉลยที่ไม่ถูกรอบงำในกรณีของการหาค่าต่ำสุด มีดังนี้



รูปที่ 2-1 แนวของพารेटอของปัญหาที่มีสองเป้าหมาย

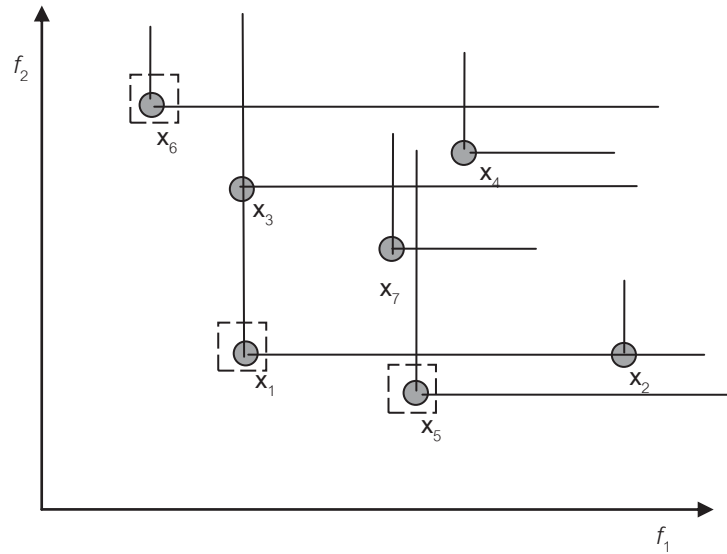
นิยาม 1 การครอบงำ (dominance) กำหนดให้ $f_i(\mathbf{x})$; $i = 1, \dots, p$ คือฟังก์ชันเป้าหมาย ถ้า $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ สำหรับทุก $i \in \{1, \dots, p\}$ และมี $j \in \{1, \dots, p\}$ อย่างน้อยหนึ่งค่า โดยที่ $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ สามารถสรุปได้ว่า $\mathbf{x}_1$ ครอบงำ $\mathbf{x}_2$ เขียนแทนด้วย $\mathbf{x}_1 < \mathbf{x}_2$.

รูปที่ 2-2 แสดงผลเฉลยในโดเมนของฟังก์ชันเป้าหมายสองฟังก์ชัน จากรูปแสดงให้เห็นว่าผลเฉลย $\mathbf{x}_1$ ครอบงำผลเฉลย $\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$ ในขณะที่ $\mathbf{x}_5$ ไม่ถูกครอบงำโดย $\mathbf{x}_1$ ในรูปยังแสดงให้เห็นถึงพื้นที่ที่ครอบงำของผลเฉลย $\mathbf{x}_1$ ซึ่งหมายความว่า ผลเฉลยใดๆ ที่อยู่ในขอบเขตพื้นที่ที่ครอบงำนี้จะถูกครอบงำโดย $\mathbf{x}_1$



รูปที่ 2-2 พื้นที่ที่ครอบงำของผลเฉลย $\mathbf{x}_1$

นิยาม 2 ผลเฉลยที่ไม่ถูกครอบงำ (non-dominated solutions) กำหนดให้ G คือเซตของผลเฉลยขนาด N , ผลเฉลย $\mathbf{x}_e \in G$ เรียกว่าผลเฉลยที่ไม่ถูกครอบงำใน G ถ้าไม่มี $\mathbf{x} \in G$ ที่ครอบงำ $\mathbf{x}_e$
จากรูปที่ 2-3 ผลเฉลยที่ไม่ถูกครอบงำคือ $\mathbf{x}_1, \mathbf{x}_5$ และ $\mathbf{x}_6$



รูปที่ 2-3 การหาผลเฉลยที่ไม่ถูกรอบงำ

ในกระบวนการคำนวณเชิงตัวเลข การหาผลเฉลยที่ไม่ถูกรอบงำสามารถทำได้โดยใช้เมตริกซ์ของการครอบงำ (dominance matrix) $\mathbf{A}$ ซึ่งแสดงวิธีการหาในตารางที่ 2-1 เอลิเมนต์ A_{ij} สามารถหาได้ดังนี้

$$A_{ij} = \begin{cases} 1; \mathbf{x}_i \prec \mathbf{x}_j \\ 0; \text{otherwise} \end{cases} \quad (2.2)$$

เมื่อได้เมตริกซ์ของการครอบงำแล้ว คำนวณหาค่าการถูกรอบงำของผลเฉลย $\mathbf{x}_j$ ได้ดังนี้

$$B_j = \sum_{i=1}^N A_{ij} \quad (2.3)$$

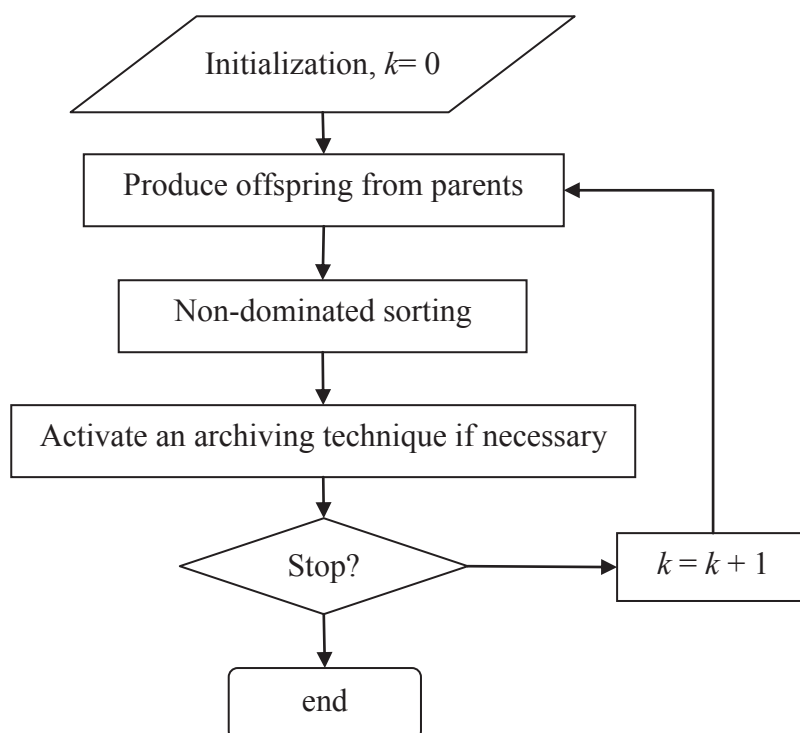
ผลเฉลย $\mathbf{x}_j$ เรียกว่าผลเฉลยที่ไม่ถูกรอบงำถ้า $B_j = 0$

ตารางที่ 2-1 เมตริกซ์สำหรับคัดสรรผลเฉลยที่ไม่ถูกรอบงำ $\mathbf{A}$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$
$\mathbf{x}_1$	0	1	1	1	0	0	1
$\mathbf{x}_2$	0	0	0	0	0	0	0
$\mathbf{x}_3$	0	0	0	1	0	0	0
$\mathbf{x}_4$	0	0	0	0	0	0	0
$\mathbf{x}_5$	0	1	0	1	0	0	0
$\mathbf{x}_6$	0	0	0	0	0	0	0
$\mathbf{x}_7$	0	0	0	1	0	0	0
Σ	0	2	1	4	0	0	1

รูปที่ 2-4 แสดงโฟลชาร์ทแบบทั่วไปของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย กระบวนการค้นหาคำตอบเริ่มต้นด้วยประชากรตั้งต้น (initial population) ซึ่งได้มาจากการสุ่มและเมตริกซ์สำหรับเก็บผลเฉลยที่ไม่ถูกรอบงำ จากนั้นทำการผลิตประชากรลูก (offspring) จากประชากรพ่อแม่ เมื่อได้ประชากรลูกแล้วทำการหาค่าฟังก์ชันเป้าหมายของผลเฉลยในประชากรลูก จากนั้นทำการคัดสรรผลเฉลยที่ไม่ถูก

ครอบงำจากยูเนียนเซตของประชากรลูกและสมาชิกในเมตริกซ์เก็บผลเฉลยที่ไม่ถูกครอบงำเดิม เมื่อได้กลุ่มของผลเฉลยที่ไม่ถูกครอบงำชุดใหม่ ไล่แทนที่ผลเฉลยที่ไม่ถูกครอบงำชุดเก่าในเมตริกซ์ ทำการปรับปรุงเมตริกซ์บรรจุผลเฉลยที่ไม่ถูกครอบงำไปเรื่อยๆจนกระทั่งหยุดการค้นหาคำตอบ ผลเฉลยพारेโตจากการใช้วิธีวิวัฒนาการเหล่านี้คือสมาชิกในเมตริกซ์บรรจุผลเฉลยที่ไม่ถูกครอบงำนั่นเอง อย่างไรก็ตามผลเฉลยนี้เรียกว่าค่าประมาณของแนวของพारेโต (approximate Pareto front) เท่านั้น เพราะไม่มีการรับประกันว่าวิธีวิวัฒนาการจะหาแนวพारेโตแน่นอนตรง (exact Pareto front) ได้



รูปที่ 2-4 โฟลชาร์ทแบบทั่วไปของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย

จากที่ได้กล่าวถึงหลักการและข้อดีของการแก้ปัญหาการออกแบบหาค่าเหมาะที่สุดด้วยขั้นตอนวิธีวิวัฒนาการมาแล้วข้างต้น ในการศึกษาครั้งนี้ได้เลือกใช้ขั้นตอนวิธีวิวัฒนาการที่ใช้ในปัจจุบันและได้รับความนิยมมาก ดังต่อไปนี้

2.2 การหาค่าเหมาะที่สุดแบบหลายเป้าหมายด้วย MPSO

MPSO ย่อมาจาก multiobjective particle swarm optimization และในกรณีของการหาค่าเหมาะที่สุดแบบเป้าหมายเดียวเราเรียกวิธีการนี้ว่าวิธี particle swarm optimization หรือ PSO ซึ่งเป็นขั้นตอนวิธีที่มีแรงดลใจมาจากธรรมชาติ โดยเฉพาะการเคลื่อนที่ของฝูงปลาหรือฝูงนก การเคลื่อนที่ทั้งสองแบบเป็นการเคลื่อนที่ขององค์ประกอบย่อย ๆ ที่เคลื่อนที่ด้วยกันอย่างประสานเวลา ปลาหรือนกสามารถเคลื่อนที่เป็นฝูง แล้วรวมตัวกลับเข้ามาฝูงอีกครั้ง ดังนั้นจึงมีการนำเอาลักษณะการเคลื่อนที่ดังกล่าวมาใช้ในการหาคำตอบของปัญหาการออกแบบหาค่าเหมาะที่สุด โดยแทนการเคลื่อนที่ของฝูงนกและฝูงปลาด้วยกลุ่มของอนุภาคซึ่งสามารถพิจารณาเป็นพฤติกรรมทางสังคมได้ สำหรับ PSO ถูกนำเสนอเป็นครั้งแรกเมื่อปี ค.ศ.1995

PSO มีคุณลักษณะหลายอย่างที่เหมือนกับการคำนวณเชิงวิวัฒนาการ เช่น ขั้นตอนวิธีทางพันธุศาสตร์ โดย PSO มีการสุ่มสร้างประชากรเริ่มต้น และใช้ในการหาคำตอบที่เหมาะสมที่สุดด้วยการปรับประชากรนั้น ๆ ในทุกรอบการคำนวณ (ทุกรุ่นหรือทุกววน) อย่างไรก็ดี PSO มีความแตกต่างไปจากขั้นตอนวิธีทางพันธุศาสตร์ตรงที่ไม่มีปฏิบัติการทางสายพันธุ์ เช่น การทำครอสโอเวอร์ (crossover) หรือมิวเตชัน (mutation) ใน PSO คำตอบของระบบถูกแทนด้วยตำแหน่งของอนุภาคที่เคลื่อนที่ไปในปริภูมิค้นหา (search space) โดยมีทิศทางไปตามตำแหน่งของอนุภาคที่ใกล้เคียงกับคำตอบที่ดีที่สุด ณ ขณะนั้น เมื่อเทียบกับขั้นตอนวิธีทางพันธุศาสตร์แล้ว ข้อได้เปรียบของ PSO คือความง่ายของขั้นตอนวิธี ในการหาคำตอบที่เหมาะสมที่สุดแบบหลายเป้าหมายนั้น ขั้นตอนการดำเนินการค้นหาเริ่มต้นจากการกำหนดกลุ่มประชากรของตำแหน่งอนุภาคเริ่มต้น พร้อมด้วยความเร็วของตำแหน่งอนุภาคเริ่มต้น หลังจากนั้นหาค่าฟังก์ชันเป้าหมายของแต่ละตำแหน่งของอนุภาคย่อยในกลุ่มประชากรของตำแหน่งอนุภาคเริ่มต้น ผลเฉลยที่ไม่ถูกรอบงำในกลุ่มประชากรของตำแหน่งอนุภาคเริ่มต้นจะถูกนำไปเก็บไว้ในเมตริกซ์เก็บผลเฉลยของพารेटโต รายละเอียดขั้นตอนการดำเนินการของ MPSO สำหรับการหาคำตอบที่เหมาะสมที่สุดแบบหลายเป้าหมายเป็นดังนี้

1: กำหนดให้หมายเลขววน (loop) $k = 0$, ทำการสุ่มค่าตำแหน่งของอนุภาคย่อย, $x_i(k)$ และความเร็วของแต่ละตำแหน่งอนุภาคย่อย $v_i(k)$ ในกลุ่มประชากรของตำแหน่งอนุภาคเริ่มต้น

2: ทำการหาค่าฟังก์ชันเป้าหมาย $f(x_i(k))$ ของแต่ละตำแหน่งของอนุภาคย่อย ซึ่งค่าฟังก์ชันเป้าหมายจะถูกพิจารณาดังนี้

2.1: ถ้าค่าของฟังก์ชันเป้าหมาย $f(x_i(k))$ ที่ได้จากตำแหน่งของอนุภาคย่อย $x_i(k)$ มีค่าดีกว่าค่าฟังก์ชันเป้าหมายทั้งหมดของกลุ่มของตำแหน่งอนุภาคย่อยทั้งหมด กำหนดให้ $g_i^{best} = x_i(k)$

2.2: ถ้าค่าของฟังก์ชันเป้าหมาย $f(x_i(k))$ ที่ได้จากตำแหน่งของอนุภาคย่อย $x_i(k)$ มีค่าดีกว่าค่าฟังก์ชันเป้าหมายที่ดีที่สุดของตำแหน่งของอนุภาคย่อยนั้น กำหนดให้ $p_i^{best} = x_i(k)$

3: หาค่าเมตริกซ์เก็บผลเฉลยภายนอกของพารेटโต A^k จากการพิจารณาค่าของ $f(x_i(k))$

4: กำหนดให้ $k = k + 1$, ทำการปรับความเร็วของตำแหน่งของอนุภาคย่อย $v_i(k)$ ตามสมการที่ (2.4) ดังนี้

$$v_i(k) = W_i v_i(k-1) + C_1 r_1 (p_i^{best} - x_i(k)) + C_2 r_2 (g_i^{best} - x_i(k)) \quad (2.4)$$

เมื่อ $x_i(k)$ คือ ตำแหน่งของอนุภาคย่อยที่ i^{th} ของรอบการคำนวณ k^{th}

$v_i(k)$ คือ เวกเตอร์ความเร็วของตำแหน่งอนุภาคย่อย $x_i(k)$ ที่รอบการคำนวณ k^{th}

W_i คือ ค่าน้ำหนักความเฉื่อยที่ใช้สำหรับควบคุมการเคลื่อนที่ของกลุ่มอนุภาค

$r_1, r_2 \in [0,1]$ คือ ตัวเลขการสุ่มแบบสม่ำเสมอ

C_1 คือ ปัจจัยการเรียนรู้เชิงปริชาณ (Cognitive learning factor)

C_2 คือ ปัจจัยการเรียนรู้ทางสังคม (Social Learning factor)

p_i^{best} คือ ค่าเหมาะสมที่สุดที่สุดของอนุภาคย่อย x_i (Personal best)

g_i^{best} คือ ค่าเหมาะสมที่สุดที่สุดวงกว้าง (Global best)

สำหรับค่าของน้ำหนักความเฉื่อย (W_i) ในสมการที่ (2.4) คำนวณค่าได้จากสมการที่ (2.5) ดังต่อไปนี้

$$W_i = W_{st} + \frac{(W_{en} - W_{st})(i)}{nloop} \quad (2.5)$$

โดยที่ W_{st} คือ ค่าน้ำหนักความเฉื่อยเริ่มต้น

W_{en} คือ ค่าน้ำหนักความเฉื่อยสุดท้าย

$nloop$ คือ จำนวนรอบของการคำนวณทั้งหมด

i คือ รอบของการคำนวณ

5: ทำการปรับค่าตำแหน่งของอนุภาคย่อย $x_i(k)$ ตามสมการที่ (2.6) ดังนี้

$$x_i(k) = x_i(k-1) + v_i(k) \quad (2.6)$$

6: หาค่าฟังก์ชันเป้าหมายจากตำแหน่งของอนุภาคย่อยที่ได้จากขั้นตอนที่ 4 แล้วพิจารณาค่า p_i^{best} และ g_i^{best} ตามขั้นตอนที่ 2

7: หาผลเฉลยภายนอกของพารโตค่าใหม่ A^k จากเมตริกซ์เก็บผลเฉลยภายนอกของพารโตรอบก่อน A^{k-1} ร่วมกับการพิจารณาจากค่าฟังก์ชันเป้าหมาย $f(x_i(k))$

8: หากเมตริกซ์เก็บผลเฉลยภายนอกของพารโตมีจำนวนผลเฉลยที่ไม่ถูกรอบงำมากกว่าค่าที่กำหนดไว้ ใช้กระบวนการกรองผลเฉลยเพื่อกำจัดผลเฉลยในส่วนที่เกินทิ้งไป

9: หยุดการทำงาน ถ้าจำนวนรอบครบตามที่กำหนดไว้ ($k = k_{max}$) หรือถ้าได้เป้าหมายตามเงื่อนไขหยุดที่กำหนดไว้

10: ถ้าจำนวนรอบยังไม่ครบตามที่กำหนด กลับไปขั้นตอนที่ 3

2.3 การหาค่าเหมาะที่สุดแบบหลายเป้าหมายด้วย SPEA2

SPEA เป็นวิธีประมาณแนวของพารโตที่ถูกพัฒนาขึ้นมาโดยมีพื้นฐานมาจากขั้นตอนวิธีทางพันธุศาสตร์ ซึ่งในตอนนี้ได้มีการพัฒนาต่อจนกลายเป็นเวอร์ชันสอง (SPEA2) สำหรับ SPEA เวอร์ชันสอง จะเริ่มต้นทำการค้นหาคำตอบจากกลุ่มของประชากรตั้งต้น โดยที่สมาชิกในเมตริกซ์เก็บผลเฉลยภายนอกของพารโตได้มาจากการจัดประเภทหรือเรียงลำดับของประชากรต้นแบบ มีการกำหนดค่าฟังก์ชันความเหมาะสม (Fitness Function) ให้กับสมาชิกเมตริกซ์ที่เกิดจากการรวมกันของประชากรในรอบปัจจุบัน และเมตริกซ์เก็บผลเฉลยภายนอกของพารโตเพื่อให้เกิดการคัดเลือกตามสิ่งแวดล้อม (Environmental Selection) โดยพิจารณาจากค่าของฟังก์ชันความเหมาะสม ในกรณีที่สมาชิกในเมตริกซ์เก็บผลเฉลยภายนอกของพารโตมีมากกว่าค่าที่กำหนดเอาไว้ให้รับได้ ผลเฉลยบางตัวจะต้องโดนกำจัดออกไปจากเมตริกซ์เก็บผลเฉลยโดยใช้ตัวดำเนินการกรองผลเฉลยจนทำให้จำนวนผลเฉลยที่อยู่ในเมตริกซ์ผลเฉลยมี

จำนวนเท่ากับค่าที่กำหนดไว้ หลังจากนั้นผลเฉลยที่ได้จากเมตริกซ์เก็บผลเฉลยจะถูกนำมาจับคู่และผ่านการเลือกแบบแข่งขัน (Tournament Selection) ผลเฉลยที่ถูกเลือกจะถูกนำมาแลกเปลี่ยนพันธุกรรมหรือการครอสโอเวอร์และเกิดการมิวเตชันเพื่อให้เกิดประชากรชุดใหม่เพื่อใช้ในการหาค่าฟังก์ชันเป้าหมายค่าใหม่ต่อไป ในการดำเนินการค้นหาคำตอบประชากรและเมตริกซ์เก็บผลเฉลยจะถูกวนรอบปรับปรุงไปเรื่อย ๆ จนกว่าจะถึงค่าจำนวนรอบหรือเงื่อนไขที่กำหนดไว้ให้หยุดทำงาน

ขั้นตอนการทำงานของSPEA เวอร์ชันสอง ในการหาค่าเหมาะที่สุดแบบหลายเป้าหมายสามารถอธิบายการทำงานของแต่ละขั้นตอนได้ดังนี้

1: เริ่มต้นการทำงาน: กำหนดให้ $k=0$, ประชากรต้นแบบที่มีขนาดเท่ากับ N คือ $P^k = \{x_1, x_2, \dots, x_N\}$, หาค่าฟังก์ชันเป้าหมาย $f(P^k)$, และกำหนดค่าสมาชิกของเมตริกซ์เก็บผลเฉลยภายนอกของพารेटอ $A^k = \{ \}$ (โดยที่จำนวนสมาชิกในเมตริกซ์เก็บผลเฉลยเท่ากับ M)

2: คำนวณค่าฟังก์ชันความเหมาะสม F^k ของสมาชิกแต่ละตัวของผลรวมของเซตประชากรต้นแบบและเมตริกซ์ผลเฉลยภายนอกของพารेटอ ($P^k \cup A^k$) เพื่อทำการคัดเลือกประชากร โดยที่ค่าฟังก์ชันความเหมาะสมสามารถหาได้จากสมการที่ (2.7) ดังนี้

$$F(i) = R(i) + D(i) \quad (2.7)$$

เมื่อ $F(i)$ คือ ฟังก์ชันความเหมาะสม

$R(i)$ คือ ฟังก์ชันความเหมาะสมหยาบ (Raw Fitness Function) หาค่าได้จากสมการที่ (2.8)

$D(i)$ คือ ฟังก์ชันความหนาแน่น (Density) หาค่าได้จากสมการที่ (2.10)

$$R(i) = \sum_{j \in (A^k \cup P^k), j \succ i} S(j) \quad (2.8)$$

โดยที่ค่าความแข็งแรง ($S(i)$) หาได้จากสมการที่ (2.9) ดังนี้

$$S(i) = |\{j | j \in A^k \cup P^k \wedge i \succ j\}| \quad (2.9)$$

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (2.10)$$

เมื่อ σ_i^k คือ ระยะทางระหว่างผลเฉลยที่ i กับผลเฉลยที่ k

$k = \sqrt{N+M}$ คือ หมายเลขของผลเฉลยอ้างอิง

3: ปรับปรุงเมตริกซ์เก็บผลเฉลี่ยของพาริโตด้วยวิธีการคัดเลือกประชากรของ $(P^k \cup A^k)$ ด้วยการคัดเลือกตามสิ่งแวดล้อม โดยใช้ค่า F^k จากขั้นตอนที่ 2 ผลที่ได้จากการคัดเลือกคือเมตริกซ์เก็บผลเฉลี่ยของพาริโตใหม่ A^{k+1} สำหรับการคัดเลือกผลเฉลี่ยในขั้นตอนนี้สามารถเขียนเป็นสมการได้ดังนี้

$$A^{k+1} = \{i | i \in A^k \cup P^k \wedge F(i) < 1\} \quad (2.11)$$

4: ถ้าจำนวนสมาชิกของ A^{k+1} มากกว่าค่าที่กำหนดไว้ให้ทำการกำจัดสมาชิกบางตัวทิ้งไป

5: คัดเลือกประชากรต้นแบบโดยใช้การเลือกด้วยการแข่งขันแบบเลขฐานสอง (Binary Tournament Selection) จาก A^{k+1} ผลที่ได้คือ เมตริกซ์ประชากรของตัวแปรออกแบบของรอบการคำนวณที่ k^{th} , $P_1^k = select(A^{k+1})$

6: ทำการแลกเปลี่ยนพันธุกรรมของประชากรของตัวแปรออกแบบใน P_1^k ผลที่ได้คือเมตริกซ์ประชากรของตัวแปรออกแบบของรอบการคำนวณที่ k^{th} , $P_2^k = recombine(P_1^k)$ โดยที่ประชากรลูกที่ได้จากการแลกเปลี่ยนพันธุกรรม เป็นไปตามสมการที่ (2.12)

$$x_i = \begin{cases} x_i^1 \text{ or } x_i^2 & \text{Pr} < 1/3 \\ x_i^1 + \lambda(x_i^2 - x_i^1) & 1/3 \leq \text{Pr} < 2/3 \\ x_i^1 + rand_1 \cdot (x_i^2 - x_i^1) & \text{Pr} \geq 2/3 \end{cases} \quad (2.12)$$

โดยที่ $\text{Pr} \in [0,1]$, $\lambda \in [-0.25, 1.25]$ และ $rand_1 \in [-0.25, 1.25]$ คือ การสุ่มตัวเลขแบบสม่ำเสมอ

7: มิวเตชัน: $P^{k+1} = mutate(P_2^k)$ โดยประชากรที่ได้จากการเกิดมิวเตชันหรือการผ่าเหล่าเป็นไปตามสมการที่ (2.13)

$$x_i = \begin{cases} l_i + rand_2 \cdot (x_i^P - l_i) & \text{Pr} < 0.5 \\ x_i^P + rand_2 \cdot (u_i - x_i^P) & \text{otherwise} \end{cases} \quad (2.13)$$

โดยที่ $rand_2 \in [-0.25, 1.25]$ คือ การสุ่มตัวเลขแบบสม่ำเสมอ

l_i คือ ขอบเขตล่างของการสุ่ม

u_i คือ ขอบเขตบนของการสุ่ม

8: หาค่าฟังก์ชันเป้าหมาย, $f(P^{k+1})$, โดยใช้ประชากรตัวแปรออกแบบที่ได้จากขั้นตอนที่ 7

9: หยุดการทำงาน ถ้าวนรอบการทำงานครบตามที่กำหนดไว้ $k = k_{\max}$, หรือถ้ายังวนรอบไม่ครบตามที่กำหนดไว้ กำหนดให้ $k = k + 1$ แล้วไปที่ขั้นตอนที่ 2

2.4 การหาค่าเหมาะที่สุดแบบหลายเป้าหมายด้วย PBIL

PBIL เป็นวิธีการหาค่าเหมาะที่สุดแบบหลายเป้าหมายโดยใช้สิ่งที่เรียกว่าเวกเตอร์ของความน่าจะเป็นในการแสดงประชากร ซึ่งอยู่ในรูปของตัวเลขฐานสอง ซึ่งความน่าจะเป็นในที่นี้คือโอกาสที่จะมีเลข “1” อยู่

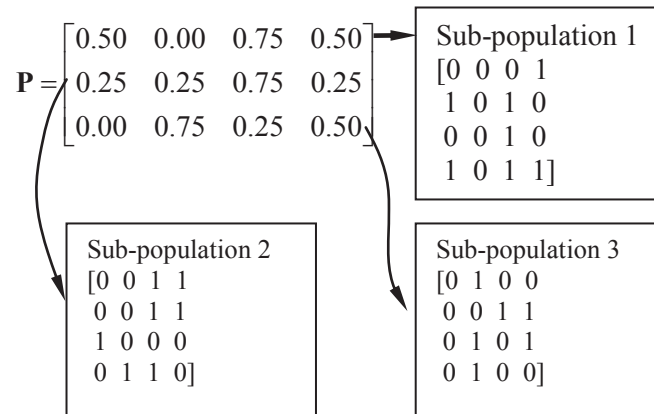
ในหลัก (Column) ที่สอดคล้องกันของประชากร ซึ่งหนึ่งเวกเตอร์ความน่าจะเป็นจะสามารถสร้างประชากรได้หลากหลายชุด ดังแสดงในรูปที่ 2-5

population 1				population 2				population 3			
0	0	1	1	1	1	0	0	0	0	1	
1	1	1	0	0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1	1	0	0	1
1	0	0	0	0	0	0	1	0	1	0	1
Probability Vectors											
[0.5, 0.5, 0.5, 0.5]				[0.5, 0.5, 0.5, 0.5]				[0.25, 0.5, 0, 0.75]			

รูปที่ 2-5 เวกเตอร์ความน่าจะเป็นและประชากรที่เกี่ยวข้อง

ในกรณีของการประยุกต์ใช้PBILกับการหาค่าเหมาะที่สุดแบบหลายเป้าหมายนั้น เพื่อให้เกิดความหลากหลายของประชากรของPBIL จำเป็นต้องใช้เวกเตอร์ความน่าจะเป็นมากกว่าหนึ่งเวกเตอร์ ดังนั้นเมื่อมีเวกเตอร์ความน่าจะเป็นมากกว่าหนึ่งเวกเตอร์จึงเรียกกลุ่มเวกเตอร์นั้นว่า เมตริกซ์ของความน่าจะเป็น ในขั้นตอนเริ่มต้น เอลิเมนต์ทุกเอลิเมนต์ของเมตริกซ์ความน่าจะเป็นจะถูกกำหนดค่าให้เป็น 0.5 ซึ่งหมายความว่ามีการกระจายของตัวเลข “0” และ “1” ในหลักให้มีจำนวนเท่ากัน กำหนดให้ N_p คือจำนวนผลเฉลยในประชากร, l คือจำนวนเวกเตอร์ของความน่าจะเป็นที่ต้องการใช้และ n_b คือความยาวของตัวเลขฐานสองสำหรับผลเฉลยใดๆ ดังนั้นเมตริกซ์ความน่าจะเป็นจะมีขนาดเป็น $l \times n_b$ โดยในแต่ละแถวของเมตริกซ์คือเวกเตอร์ความน่าจะเป็นย่อยที่ใช้ในการผลิตประชากรย่อยเป็นจำนวน N_p/l ผลเฉลย ตัวแปรออกแบบแต่ละตัวแปรถูกแสดงด้วยตัวเลขฐานสองด้วยจำนวน n_b/n เอลิเมนต์ เมื่อ n คือจำนวนของตัวแปรออกแบบทั้งหมด

การผลิตประชากรในแต่ละรอบของการคำนวณ ทำได้จากการรวมกลุ่มประชากรย่อย l กลุ่มเข้าด้วยกัน (แสดงในรูปที่ 2-6) จากนั้นทำการหาค่าฟังก์ชันเป้าหมายของผลเฉลยในกลุ่มประชากรที่สร้างขึ้นมา ขั้นตอนต่อมาคือการคัดสรรผลเฉลยที่ไม่ถูกรอบงำจากการรวมกลุ่มของประชากรชุดใหม่ที่ได้กับผลเฉลยที่ไม่ถูกรอบงำในรอบการคำนวณที่ผ่านมา ซึ่งในระหว่างรอบการคำนวณของวิธีวิวัฒนาการส่วนใหญ่สามารถค้นหาผลเฉลยที่ไม่ถูกรอบงำได้เป็นจำนวนมาก ดังนั้นถ้าจำนวนผลเฉลยที่ไม่ถูกรอบงำมีจำนวนมากกว่าค่าที่เรากำหนดเอาไว้ (กำหนดให้ N_A คือจำนวนผลเฉลยที่ไม่ถูกรอบงำ) เราสามารถแก้ปัญหานี้ได้โดยใช้กระบวนการคำนวณเชิงตัวเลขที่เรียกว่าวิธีการกรองผลเฉลยสำหรับเก็บไว้ในเมตริกซ์ผลเฉลย ในแต่ละรอบของการคำนวณ เมตริกซ์ความน่าจะเป็นจะถูกปรับปรุงโดยใช้ผลเฉลยเลขฐานสองที่สุ่มมาจากเมตริกซ์เก็บผลเฉลยของพารามิเตอร์ในปัจจุบันเพื่อใช้ในการปรับปรุงแถวที่ i^{th} ของเมตริกซ์ความน่าจะเป็น ดังนั้นแถวที่ i^{th} ของเมตริกซ์ความน่าจะเป็นจะถูกปรับปรุง กระบวนการปรับปรุงจะสำเร็จก็ต่อเมื่อทุกแถวของเมตริกซ์ความน่าจะเป็นถูกเปลี่ยนแปลง เมตริกซ์ความน่าจะเป็นถูกปรับปรุงและเมตริกซ์เก็บผลเฉลยภายนอกของพารามิเตอร์ถูกทำให้ดีขึ้นด้วยการวนซ้ำจนกระทั่งพบเงื่อนไขการหยุดทำงาน



รูปที่ 2-6 แสดงตัวอย่างของเมตริกซ์ความน่าจะเป็นและประชากรที่เกี่ยวข้อง

สำหรับขั้นตอนการทำงานของPBILแบบหลายเป้าหมายสามารถเขียนอธิบายได้ดังนี้

- 1: เริ่มการทำงาน โดยกำหนดให้เมตริกซ์ความน่าจะเป็นมีค่าเท่ากับ $P = [0.5]_{L \times n_b}$ และเมตริกซ์เก็บผลเฉลยภายนอกของพารेटอ $Pareto = \{ \}$
- 2: สร้างเมตริกซ์ของประชากรเลขฐานสอง B (ตัวแปรออกแบบที่เป็นเลขฐานสอง) จากเมตริกซ์ความน่าจะเป็น P
- 3: ถอดรหัสประชากรเลขฐานสองให้กลายเป็นตัวแปรออกแบบที่เป็นเลขจำนวนจริง $x_{n \times N_p}$ และหาค่าฟังก์ชันเป้าหมาย $f_{m \times N_p}$
- 4: ปรับปรุงเมตริกซ์เก็บผลเฉลยภายนอกของพารेटอ $Pareto$ โดยแทนที่ด้วยผลเฉลยที่ไม่ถูกรอบงำของการรวมตัวของ $Pareto$ และ x ($Pareto \cup x$)
- 5: ถ้าจำนวนสมาชิกของเมตริกซ์เก็บผลเฉลยภายนอกของพารेटอเกินกว่าจำนวนสมาชิกที่กำหนดไว้ N_A ลบสมาชิกบางตัวทิ้งโดยใช้วิธีการของผลเฉลย
- 6: ถ้าตรงตามเงื่อนไขหยุดทำงานให้หยุดทำงาน แต่ถ้าไม่ตรงให้ไปทำงานต่อในข้อ 7
- 7: ปรับปรุงเมตริกซ์ความน่าจะเป็น P โดยมีขั้นตอนดังนี้
 - 7.1: For $i=1$ to L
 - 7.2: เลือกจำนวนผลเฉลยเลขฐานสอง n_0 ด้วยการสุ่มจากเมตริกซ์เก็บผลเฉลยภายนอกของพารेटอ $Pareto$
 - 7.3: ปรับปรุงแถวที่ i^{th} ของเมตริกซ์ความน่าจะเป็น P ตามสมการที่ (2.14)

$$P_{i,j}^{new} = P_{i,j}^{old} (1 - L_R) + \bar{b}_j L_R \quad (2.14)$$

เมื่อ L_R คือ อัตราการเรียนรู้ หาได้จากสมการที่ (2.15)

b_j คือ ค่าเฉลี่ยของหลักที่ j^{th} ของผลเฉลยเลขฐานสองที่ถูกสุ่มมาปรับปรุงเมตริกซ์ความน่าจะเป็น

$$L_R = 0.5 + rand \cdot (+0.1 \text{ or } -0.1) \quad (2.15)$$

7.4: พิจารณาการเกิดมิวเตชันของเมตริกซ์ความน่าจะเป็น P ตามสมการที่ (2.16)

$$P_{i,j}^{new} = P_{i,j}^{old} (1 - m_s) + rand(0 \text{ or } 1) \cdot m_s \quad (2.16)$$

โดยที่ $rand \in [0,1]$ คือ การสุ่มแบบสม่ำเสมอ

m_s คือ ค่าการเลื่อนของมิวเตชัน

7.5: Next i

8: กลับไปที่ขั้นตอนที่ 2

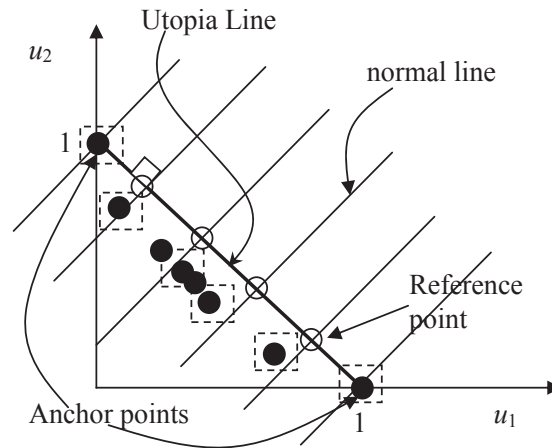
ขั้นตอนการกรองผลเฉลยที่ไม่ถูกรอบง่าที่เกินกว่าเมตริกซ์เก็บผลเฉลยภายนอกของพารโตจะรับได้ สำหรับวิธี PBIL ใช้เทคนิควิธีเส้นตั้งฉาก (Normal Line Technique) ในการกรองผลเฉลย สำหรับขั้นตอนการทำงานของวิธีการเส้นตั้งฉาก มีดังต่อไปนี้ เริ่มต้นผลเฉลยปัจจุบันที่ไม่ถูกรอบง่าทั้งหมดจะถูกนำมาทำให้เป็นบรรทัดฐาน (Normalize) ตามสมการที่ (2.17)

$$u_i = \frac{(f_i - f_i^{\min})}{(f_i^{\max} - f_i^{\min})} \quad (2.17)$$

เมื่อ $f^{\min}$ คือ ค่าต่ำสุดของฟังก์ชันเป้าหมายในรอบที่ i^{th} หาได้จากผลเฉลยที่ไม่ถูกรอบง่าในปัจจุบัน

$f^{\max}$ คือ ค่าสูงสุดของฟังก์ชันเป้าหมายในรอบที่ i^{th} หาได้จากผลเฉลยที่ไม่ถูกรอบง่าในปัจจุบัน

ขั้นตอนกระบวนการเชิงตัวเลขของเทคนิควิธีเส้นตั้งฉากในกรณีการหาค่าเหมาะสมสุดแบบสองเป้าหมายแสดงไว้ในรูปที่ 2-7 จุดที่ให้ค่าต่ำสุดของแต่ละฟังก์ชันเป้าหมายเรียกว่าจุดทอดสมอและเส้นที่เชื่อมระหว่างจุดทั้งสองเรียกว่าเส้นยูโทเปีย จากจุดทอดสมอและเส้นยูโทเปียในรูปที่ 2-7 ทำการแบ่งจุดอ้างอิงบนเส้นยูโทเปียออกเป็นจำนวน N_a จุด โดยให้ระยะห่างระหว่างจุดมีค่าเท่า ๆ กัน เส้นตั้งฉากในที่นี้คือเส้นที่ลากผ่านจุดอ้างอิงและตั้งฉากกับเส้นยูโทเปีย ผลเฉลยที่จะถูกเก็บในเมตริกซ์เก็บผลเฉลยภายนอกของพารโตคือผลเฉลยที่มีค่าอยู่ใกล้กับเส้นตั้งฉากมากที่สุด (แสดงไว้ด้วยกรอบเส้นประสีเหลี่ยมในรูปที่ 2-7) รายละเอียดเพิ่มเติมอยู่ใน [18]



รูปที่ 2-7 แสดงเทคนิควิธีเส้นตั้งฉาก

2.5 การหาค่าเหมาะที่สุดแบบหลายเป้าหมายด้วย NSGA-II

NSGA-II นำเสนอโดย Kalyanmoy Deb ซึ่งพัฒนาจากขั้นตอนวิธีเดิมคือ NSGA [79] ความแตกต่างคือ NSGA-II ไม่ต้องการ niching parameter (σ_{share}) ซึ่งเป็นข้อด้อยของอัลกอริทึมแบบวิวัฒนาการหลายเป้าหมายที่ใช้พารามิเตอร์นี้ (ดังใน NSGA) และถูกแทนที่ด้วย crowded-comparison approach ซึ่งให้ความหลากหลายของผลเฉลยมากกว่า วิธี NSGA-II แตกต่างจากวิธีที่กล่าวมาข้างต้นอย่างหนึ่งคือ วิธีนี้ไม่ใช่เมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอบงำ แต่จะทำการเก็บผลเฉลยที่ไม่ถูกรอบงำไว้ในเมตริกซ์ประชากรโดยตรง การดำเนินการของ NSGA-II กับ การหาค่าเหมาะที่สุดหลายเป้าหมาย มีรายละเอียดดังนี้

- 1: เริ่มต้นด้วย กำหนดให้ $t = 0$ กำเนิดประชากรเริ่มต้น P_t โดยมีจำนวนประชากรเท่ากับ N หาค่าฟังก์ชันเป้าหมาย $f(P_t)$
- 2: ขั้นตอนคัดเลือกของ NSGA-II

2.1: กำหนดแนวระดับ (rank assignment) หรือแนวระดับที่ไม่ถูกรอบงำ (non-dominated sorting) ของแต่ละประชากร โดยที่แนวระดับสูงสุดของผลเฉลยที่ไม่ถูกรอบงำ จะเท่ากับระดับที่ 1 และนำออกจาก P_t จากนั้นพิจารณาชุดประชากรที่เหลือและดำเนินการเช่นเดียวกัน จนกระทั่งได้แนวระดับที่ไม่ถูกรอบงำสุดท้าย

2.2: หาระยะห่างระหว่างผลเฉลยใดๆ (crowding distance) (ซึ่งเป็นตัวระบุถึงความหลากหลายของแต่ละผลเฉลย) ของกลุ่มประชากรแต่ละระดับที่ไม่ถูกรอบงำ (rank) ใน P_t

2.3: เลือกประชากรต้นกำเนิด (parent population, A_t) จาก P_t โดยการเลือกแบบไบนารี (binary selection) ซึ่งพิจารณาจากระดับที่ไม่ถูกรอบงำและระยะห่างระหว่างผลเฉลย โดยในแต่ละครั้งของการเลือก ผลเฉลย 2 ตัวจะถูกเลือกแบบสุ่ม

- กรณีที่ระดับที่ไม่ถูกรอบงำของผลเฉลยไม่เท่ากัน ผลเฉลยที่ระดับที่ไม่ถูกรอบงำสูงกว่าจะถูกเลือก
- กรณีที่ระดับที่ไม่ถูกรอบงำของผลเฉลยเท่ากัน ผลเฉลยที่อยู่ในที่หนาแน่นน้อยกว่าจะถูกเลือก

3: ดำเนินการแลกเปลี่ยนพันธุกรรม (crossover) และการผ่าเหล่า (mutation) เพื่อกำเนิดประชากรใหม่ (offspring population, Q_t)

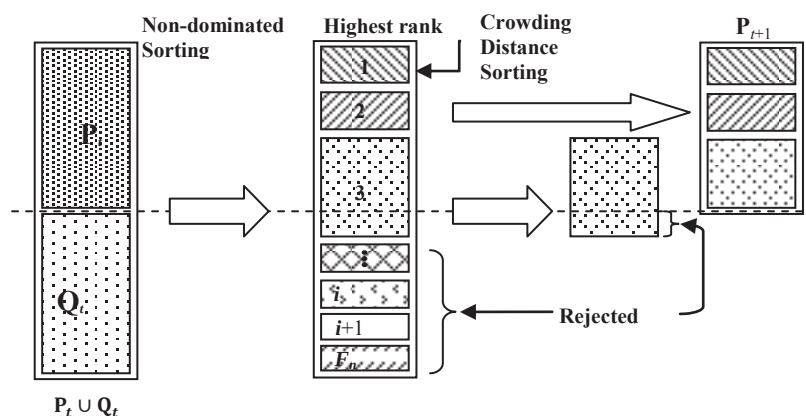
4: รวมเซตประชากรต้นกำเนิดและประชากรเกิดใหม่ $R_t = P_t \cup Q_t$ ซึ่งมีจำนวนประชากรเท่ากับ $2N$

จากนั้นกำหนดระดับที่ไม่ถูกรอบงำให้กับแต่ละผลเฉลยและหาค่าระยะห่างระหว่างผลเฉลยใดๆ ใน R_t

5: ได้ประชากรใหม่ P_{t+1} ขนาด N จาก R_t ซึ่งมีขนาด $2N$ จากขั้นตอนการคัดเลือกของ NSGA-II ดังรูปที่ 2-8

6: หยุดกระบวนการทำงาน เมื่อบรรลุตามเงื่อนไขหรือรอบการคำนวณครบตามที่กำหนดไว้ ($t = t_{\max}$) หรือมิฉะนั้น กลับไปเริ่มดำเนินการ ขั้นตอนที่ 2

7: หาผลเฉลยพारेโต **Pareto** จากประชากรชุดสุดท้าย



รูปที่ 2-8 รูปแบบการคัดเลือกของ NSGA-II

การเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการด้วยแบบจำลองทดแทน

3.1 ความนำ

แบบจำลองทดแทน (surrogate models) คือการประมาณค่าฟังก์ชันเป้าหมายหรือเงื่อนไขบังคับของปัญหาการหาค่าเหมาะที่สุด ด้วยฟังก์ชันอย่างง่ายหรือฟังก์ชันที่ใช้เวลาในการคำนวณน้อย โดยการประมาณค่าทำได้จากการสุ่มผลเฉลยของปัญหาการหาค่าเหมาะที่สุด จากนั้นทำการหาค่าฟังก์ชันจริงของผลเฉลยเหล่านั้น เมื่อทราบตำแหน่งจุดของผลเฉลยและค่าฟังก์ชัน เราสามารถประยุกต์ใช้วิธีการประมาณค่าในช่วงหรืออินเตอร์โพลชัน (interpolation) หรือวิธีกำลังสองน้อยสุด (least square method) เพื่อสร้างแบบจำลองทดแทนในรูปแบบต่างๆ ได้ เนื้อหาในบทนี้จะกล่าวถึงชนิดของแบบจำลองทดแทนที่นิยมใช้ และแนวทางในการเชื่อมต่อระหว่างแบบจำลองเหล่านั้นกับขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย

3.2 แบบจำลองทดแทน

กำหนดให้ $y = f(\mathbf{x})$ คือฟังก์ชันหรือตัวแปรตามของเวกเตอร์ออกแบบ $\mathbf{x}$ ซึ่งมีขนาดเป็น $n \times 1$ กำหนดให้เซตของกลุ่มเวกเตอร์ตัวแปรออกแบบที่ทำการสุ่มขึ้นมาคือ $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^N]$ และค่าฟังก์ชันของเวกเตอร์เหล่านั้นคือ $\mathbf{Y} = [y_1, \dots, y_N]$ การสร้างแบบจำลองทดแทนสามารถทำได้ดังนี้

3.2.1 การสร้างพื้นผิวตอบสนองแบบฟังก์ชันพหุนาม

ฟังก์ชันพหุนามที่นิยมใช้ในการสร้างแบบจำลองทดแทนในการหาค่าเหมาะที่สุดด้วยขั้นตอนวิธีวิวัฒนาการคือฟังก์ชันกำลังสอง (quadratic function) ในกรณีที่ตัวแปรต้นอยู่ในปริภูมิ n มิติ ฟังก์ชันกำลังสองสามารถเขียนได้ดังนี้

$$\bar{y} = \beta_0 + \sum \beta_i x_i + \sum \beta_{ij} x_i x_j \quad (3.1)$$

เมื่อ $\beta_i; i = 0, \dots, (n+1)(n+2)/2$ คือสัมประสิทธิ์ของฟังก์ชันกำลังสองที่ต้องทำการหาค่าในกระบวนการสร้างแบบจำลองทดแทน การหาค่าสัมประสิทธิ์ของฟังก์ชันกำลังสองสามารถทำได้โดยการใช้อินเวอร์สเมตริกซ์เทียม (pseudo-inverse) หรือวิธีกำลังสองน้อยสุด

3.2.2 แบบจำลอง Kriging

หนึ่งในการประยุกต์ใช้แบบจำลอง Kriging หรือแบบจำลองกระบวนการเกาส์เซียน (Gaussian process model) คือ MATLAB toolbox ที่อยู่ในชื่อของ DACE ซึ่งย่อมาจาก design and analysis of computer experiments [58] การประมาณค่าฟังก์ชันสามารถทำได้จากการเชื่อมโยงระหว่างการประมาณค่าแบบเฉพาะที่ (local) และแบบวงกว้าง (global) ดังนี้

$$y(\mathbf{x}) = \bar{f}(\mathbf{x}) + Z(\mathbf{x}) \quad (3.2)$$

เมื่อ $\bar{f}(\mathbf{x})$ สมการถดถอยวงกว้าง (global regression model) และ $Z(\mathbf{x})$ คือฟังก์ชันกระบวนการสโตแคสติกเกาส์เซียน (stochastic Gaussian process) ที่มีค่าเฉลี่ยเป็นศูนย์และมีความแปรปรวนร่วมเกี่ยว (covariance) ที่ไม่เป็นศูนย์ ทั้งนี้ฟังก์ชันที่สองจัดว่าเป็นการประมาณค่าเฉพาะที่เราสามารถใช้สมการเชิงเส้นแสดงสมการถดถอยวงกว้างได้ดังนี้

$$\bar{f} = \beta_0 + \sum_{i=1}^n \beta_i x_i = \boldsymbol{\beta}^T \mathbf{f} \quad (3.3)$$

เมื่อ $\boldsymbol{\beta} = \{\beta_0, \dots, \beta_n\}^T$ และ $\mathbf{f} = \mathbf{f}(\mathbf{x}) = \{1, x_1, x_2, \dots, x_n\}^T$ ค่าความแปรปรวนร่วมของ $Z(\mathbf{x})$ สามารถแสดงได้ดังนี้

$$\text{Cov}(Z(\mathbf{x}^p), Z(\mathbf{x}^q)) = \sigma^2 \mathbf{R}[\mathbf{R}(\mathbf{x}^p, \mathbf{x}^q)] \quad (3.4)$$

สำหรับ $p, q = 1, \dots, N$ เมื่อ R คือฟังก์ชันสหสัมพันธ์ (correlation) ใดๆจากจุดผลเฉลย N ที่สุ่มขึ้นมา และ $\mathbf{R}$ คือเมตริกซ์สหสัมพันธ์สมมาตรขนาด $N \times N$ ที่มีสมาชิกแนวเส้นทแยงเป็นหนึ่ง ในที่นี้ฟังก์ชันสหสัมพันธ์คือ

$$R(\mathbf{x}^p, \mathbf{x}^q) = \exp(-(\mathbf{x}^p - \mathbf{x}^q)^T \boldsymbol{\theta}(\mathbf{x}^p - \mathbf{x}^q)) \quad (3.5)$$

เมื่อ $\boldsymbol{\theta}_i$ คือพารามิเตอร์สหสัมพันธ์ที่ไม่ทราบค่าซึ่งสามารถหาได้จากวิธีค่าน่าจะเป็นสูงสุด (maximum likelihood method) เมื่อทราบค่าของ $\boldsymbol{\beta}$ และ $\boldsymbol{\theta}$ การทำนายค่าด้วยแบบจำลอง Kriging สามารถหาได้จาก

$$\bar{y} = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \mathbf{r}^T(\mathbf{x}) \mathbf{R}^{-1}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \quad (3.6)$$

เมื่อ $\mathbf{F} = [\mathbf{f}(\mathbf{x}^1), \mathbf{f}(\mathbf{x}^2), \dots, \mathbf{f}(\mathbf{x}^N)]^T$ และ $\mathbf{r}^T(\mathbf{x}) = [R(\mathbf{x}, \mathbf{x}^1), R(\mathbf{x}, \mathbf{x}^2), \dots, R(\mathbf{x}, \mathbf{x}^N)]$ สามารถดูรายละเอียดเพิ่มเติมได้ใน [58]

3.2.3 การประมาณในช่วงด้วยฟังก์ชันฐานหลักแนวรัศมี (radial basis function)

การประมาณในช่วงด้วยฟังก์ชันฐานหลักแนวรัศมีหรือบางทีเรียกว่าการประมาณค่าในช่วงด้วยพื้นผิวสไปไลน์ (surface spline interpolation) [38] เป็นการประมาณค่าในช่วงที่นิยมใช้ในงานจริงหลากหลายสาขาวิชา เช่น ใช้ในการเชื่อมต่อระหว่างแบบจำลองไฟไนต์เอลิเมนต์และแบบจำลองอากาศพลวัตในการวิเคราะห์ทางแอโรไดนามิก [81] ฟังก์ชันฐานหลักแนวรัศมีสามารถแสดงได้ดังนี้

$$\bar{y}(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\|\mathbf{x} - \mathbf{x}_i\|) \quad (3.7)$$

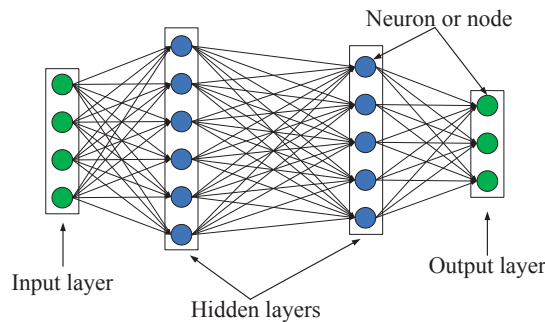
เมื่อ α_i คือสัมประสิทธิ์ของฟังก์ชันที่ต้องทำการหาค่า และ K คือฟังก์ชันฐานหลักซึ่งมีหลากหลายรูปแบบ ขึ้นอยู่กับผู้ใช้งาน จากผลเฉลยเริ่มต้น N ชุด สมการ (3.7) สามารถแสดงได้ N สมการดังนี้

$$y(\mathbf{x}_i) = \sum_{j=1}^N K(\|\mathbf{x}_i - \mathbf{x}_j\|) \alpha_j; i = 1, \dots, N \quad (3.8)$$

ซึ่งจะได้ระบบสมการเชิงเส้น N สมการและมีตัวแปรที่ไม่ทราบค่าคือ α_i N ตัว เมื่อทำการหาสัมประสิทธิ์ α_i แล้ว เราสามารถทำนายค่าฟังก์ชันจากแบบจำลองทดแทนที่จุด $\mathbf{x}$ ใดๆ ด้วยสมการ (3.7)

3.2.4 โครงข่ายประสาทเทียม (artificial neural network)

โครงข่ายประสาทเทียม ถูกพัฒนาขึ้นมาเพื่อสามารถกำหนดฟังก์ชันสำหรับแบบจำลองทดแทนที่มีความสลับซับซ้อนมากกว่าฟังก์ชันกำลังสอง หรือฟังก์ชันฐานหลักแนวรัศมี แบบจำลองโครงข่ายประสาทเทียมมีหลากหลายรูปแบบ เพราะได้รับการพัฒนามาเพื่อประยุกต์ใช้ในงานที่แตกต่างกันออกไป โดยหลักๆ แล้ว แบบจำลองโครงข่ายประสาทจะประกอบไปด้วย หน่วยประสาท (neurons) หรือจุดต่อ (nodes) ที่เชื่อมต่อกันเป็นโครงข่ายเลียนแบบระบบประสาทของสิ่งมีชีวิต รูปที่ 3-1 แสดงตัวอย่างโครงข่ายประสาทแบบ feedforward multilayer perceptrons ที่ประกอบด้วยชั้นข้อมูลขาเข้า (input) ชั้นตรงกลางหรือชั้นที่ซ่อนเอาไว้ (hidden layers) และชั้นขาออก (output layer)



รูปที่ 3-1 โครงข่ายประสาทเทียม

3.3 การผสมผสานขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายกับแบบจำลองทดแทน

การผสมผสานระหว่างขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายกับแบบจำลองทดแทนมีหลักการอยู่ว่า ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายถูกใช้เป็นตัวดำเนินการหลักในกระบวนการค้นหาผลเฉลยพาเรโต เนื่องจากวิธีหาค่าเหมาะที่สุดเหล่านี้ ใช้กลุ่มของผลเฉลยหรือที่เรียกว่าประชากรในการหาผลเฉลยเหมาะที่สุด ดังนั้นในแต่ละรอบของการคำนวณหรือวงวน (loop) วิธีการเหล่านี้จะมีเซตของผลเฉลยที่สร้างขึ้นมา ทั้งจากการสุ่มและการดำเนินการเชิงวิวัฒนาการ (evolutionary operations) ฉะนั้น เราสามารถใช้กลุ่มผลเฉลยเหล่านั้น ในการสร้างแบบจำลองทดแทนรูปแบบใดรูปแบบหนึ่ง และสามารถหาค่าเหมาะที่สุดด้วยการใช้แบบจำลองทดแทนนั้น ซึ่งจะทำให้เวลาที่ใช้ในการหาคำตอบเหมาะที่สุดสั้นลงอย่างมากเมื่อเทียบกับการหาค่าฟังก์ชันจริง อย่างไรก็ตามความแม่นยำของแบบจำลองทดแทนมีผลต่อกระบวนการหาผลเฉลยรูปแบบนี้มาก การผสมผสานระหว่างขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายรูปแบบต่างๆ มีดังต่อไปนี้

3.3.1 การใช้แบบจำลองทดแทนในขั้นตอนเริ่มต้นเท่านั้น

การผสมผสานระหว่างแบบจำลองทดแทนและขั้นตอนวิธีวิวัฒนาการในรูปแบบนี้ ใช้ในกรณีที่การหาค่าฟังก์ชันในการออกแบบมีปริมาณจำกัด หรือใช้เวลาในการวิเคราะห์ทางไฟไนต์เอลิเมนต์แต่ละครั้งนานมาก จนไม่สามารถรันกระบวนการหาค่าที่เหมาะสมที่สุดได้ แนวทางในการดำเนินการหาค่าเหมาะสมที่สุดตามแผนการนี้ทำได้โดยการใช้วิธีการสุ่มเซตของผลเฉลยตัวแปรออกแบบจำนวนหนึ่งขึ้นมา วิธีที่นิยมใช้มากที่สุดคือวิธีการสุ่มแบบลาตินไฮเปอร์คิวบ์ (Latin hypercube sampling (LHS) technique) หรืออาจใช้วิธีแบบสุ่มหรือวิธีที่เหมาะสมอื่นๆ เช่น การออกแบบเชิงตัวประกอบ (factorial design) ขั้นตอนวิธีการผสมมีดังนี้

- 1: ขั้นตอนเริ่มต้น สุ่มเวกเตอร์ออกแบบ N จุด ด้วยวิธี LHS หรือวิธีอื่นๆตามความเหมาะสม
- 2: สร้างแบบจำลองทดแทน
- 3: สร้างประชากรเริ่มต้นสำหรับขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย กำหนดเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอรับ
- 4: หาค่าฟังก์ชันของสมาชิกในกลุ่มประชกรนั้น โดยใช้แบบจำลองทดแทน
- 5: ถ้าเงื่อนไขให้หยุดการหาผลเฉลย ข้ามไปที่ขั้นตอนที่ 10 ถ้าไม่ใช่ ให้ดำเนินการในขั้นตอนต่อไป
- 6: ทำการสร้างประชากรลูก และหาค่าฟังก์ชันของประชากรเหล่านั้น
- 7: หาผลเฉลยที่ไม่ถูกรอรับของยูเนียนเซตของประชากรลูกและสมาชิกในเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอรับในรอบของการคำนวณก่อนหน้านี้
- 8: ถ้าขนาดของเมตริกซ์เก็บผลเฉลยใหญ่เกินไป คัดผลเฉลยบางตัวออกจากเมตริกซ์
- 9: กลับไปขั้นตอนที่ 5
- 10: หาค่าฟังก์ชันจริงของผลเฉลยที่ไม่ถูกรอรับจากขั้นตอนที่ 8 พร้อมทั้งหาผลเฉลยที่ไม่ถูกรอรับจริงหยุดการคำนวณ

3.3.2 การใช้แบบจำลองทดแทนในขั้นตอนเริ่มต้นและในช่วงท้ายของกระบวนการหาค่าที่เหมาะสมที่สุด

แผนการผสมผสานรูปแบบนี้ มีแนวคิดที่ว่า เราสามารถใช้แบบจำลองทดแทนในขั้นตอนเริ่มต้น หลังจากนั้นหาผลเฉลยที่เหมาะสมที่สุดด้วยแบบจำลองทดแทนนี้ จนกระทั่งได้คำตอบที่ดี จากนั้นกลับมาใช้การหาฟังก์ชันจริงในกระบวนการหาค่าตอบช่วงสุดท้ายของขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย การดำเนินการด้วยการหาค่าฟังก์ชันจริงจะช่วยให้ได้ผลเฉลยเหมาะสมที่สุดของพารามิเตอร์ที่ดีกว่าแผนการ 3.3.1 กระบวนการคำนวณมีรายละเอียดดังนี้

- 1: ขั้นตอนเริ่มต้น สุ่มเวกเตอร์ออกแบบ N จุด ด้วยวิธี LHS หรือวิธีอื่นๆตามความเหมาะสม
- 2: สร้างแบบจำลองทดแทน
- 3: สร้างประชากรเริ่มต้นสำหรับขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย กำหนดเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอรับ

- 4: หาค่าฟังก์ชันของสมาชิกในกลุ่มประชากรนั้น โดยใช้แบบจำลองทดแทน
- 5: ถ้าเงื่อนไขให้หยุดการหาผลเฉลย ข้ามไปที่ขั้นตอนที่ 11 ถ้าไม่ใช่ ให้ดำเนินการในขั้นตอนต่อไป
- 6: ทำการสร้างประชากรลูก
- 7: ถ้าวรอบของการคำนวณอยู่ในช่วงท้ายของกระบวนการ หาค่าฟังก์ชันจริงของประชากรลูกเหล่านั้น ถ้าไม่ใช่ ให้หาค่าฟังก์ชันจากแบบจำลองทดแทน
- 8: หาผลเฉลยที่ไม่ถูกรอบจำของยูเนียนเซตของประชากรลูกและสมาชิกในเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอบจำในรอบของการคำนวณก่อนหน้านี้
- 9: ถ้าขนาดของเมตริกซ์เก็บผลเฉลยใหญ่เกินไป คัดผลเฉลยบางตัวออกจากเมตริกซ์
- 10: กลับไปขั้นตอนที่ 5
- 11: หยุดการหาคำตอบ

3.3.3 การใช้แบบจำลองทดแทนในขั้นตอนเริ่มต้นและตลอดกระบวนการหาค่าที่เหมาะสมที่สุด

แผนการผสมผสานแบบนี้ อาจเรียกว่าการดำเนินการแบบคู่ขนาน กล่าวคือ เริ่มต้นด้วยการสุ่มเลือกเซตของเวกเตอร์ตัวแปรออกแบบจำนวนหนึ่งด้วยวิธีการที่เหมาะสม จากนั้นหาค่าฟังก์ชันจริงของผลเฉลยเหล่านั้นพร้อมสร้างแบบจำลองทดแทน ทำการหาผลเฉลยที่เหมาะสมที่สุดโดยใช้ฟังก์ชันทดแทน และนำผลเฉลยจากขั้นตอนนี้มาหาค่าฟังก์ชันจริง เราอาจกล่าวได้ว่า การหาค่าที่เหมาะสมที่สุดโดยใช้แบบจำลองทดแทนในขั้นตอนนี้เป็นส่วนหนึ่งของการผลิตประชากรลูกของกระบวนการหาค่าที่เหมาะสมที่สุดหลัก รายละเอียดของกระบวนการคำนวณแสดงได้ดังต่อไปนี้

- 1: ขั้นตอนเริ่มต้น สุ่มเวกเตอร์ออกแบบ N จุด ด้วยวิธี LHS หรือวิธีอื่นๆตามความเหมาะสม กำหนดเมตริกซ์เก็บผลเฉลย
- 2: ถ้าเงื่อนไขให้หยุดการหาผลเฉลย ข้ามไปที่ขั้นตอนที่ 8 ถ้าไม่ใช่ ให้ดำเนินการในขั้นตอนต่อไป
- 3: ดำเนินการหาผลเฉลยที่ไม่ถูกรอบจำด้วยกระบวนการคำนวณในหัวข้อย่อย 3.3.1
- 4: ทำการสร้างประชากรลูก และหาค่าฟังก์ชันจริงของประชากรเหล่านั้น
- 5: หาผลเฉลยที่ไม่ถูกรอบจำของยูเนียนเซตของประชากรลูกในข้อ 3 และ 4 และสมาชิกในเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอบจำในรอบของการคำนวณก่อนหน้านี้
- 6: ถ้าขนาดของเมตริกซ์เก็บผลเฉลยใหญ่เกินไป คัดผลเฉลยบางตัวออกจากเมตริกซ์
- 7: กลับไปขั้นตอนที่ 2
- 8: หยุดการค้นหาค่าผลเฉลย

3.3.4 การใช้แบบจำลองทดแทนมากกว่าหนึ่งชุดพร้อมกันในกระบวนการหาผลเฉลย

แนวคิดสำหรับการผสมผสานในรูปแบบนี้จะมีรูปแบบเดียวกันกับรูปแบบในหัวข้อที่ 3.3.3 แต่มีข้อแตกต่างคือ การใช้แบบจำลองทดแทนหลายชนิดในการผลิตประชากรลูกในขั้นตอนที่ 4 ของหัวข้อย่อย

3.3.3 สาเหตุที่ต้องใช้แบบจำลองทดแทนหลากหลายชนิดเพราะว่า แบบจำลองทดแทนแต่ละชนิด มีข้อดีข้อเสียแตกต่างกันไป ในการออกแบบแต่ละครั้งผู้ออกแบบอาจไม่ทราบว่าแบบจำลองทดแทนชนิดไหนเหมาะสมที่สุดกับปัญหาการออกแบบนั้นๆ ดังนั้นการใช้แบบจำลองทดแทนหลากหลายรูปแบบย่อมส่งผลดีกว่าการเลือกใช้แบบใดแบบหนึ่ง อย่างไรก็ตาม หากผู้ออกแบบมีองค์ความรู้แล้วว่าแบบจำลองชนิดไหนมีประสิทธิภาพดีที่สุดกับปัญหาการออกแบบที่กำลังพิจารณา ผู้ออกแบบควรเลือกใช้แบบจำลองทดแทนชนิดนั้น เพื่อความได้เปรียบในการค้นหาผลเฉลย กระบวนการคำนวณของแผนการผสมผสานรูปแบบนี้มีดังนี้

- 1: ขั้นตอนเริ่มต้น สุ่มเวกเตอร์ออกแบบ N จุด ด้วยวิธี LHS หรือวิธีอื่นๆตามความเหมาะสม กำหนดเมตริกซ์เก็บผลเฉลย
- 2: ถ้าเงื่อนไขให้หยุดการหาผลเฉลย ข้ามไปที่ขั้นตอนที่ 8 ถ้าไม่ใช่ ให้ดำเนินการในขั้นตอนต่อไป
- 3: ดำเนินการหาผลเฉลยที่ไม่ถูกรอบจำกัดด้วยกระบวนการคำนวณในหัวข้อย่อย 3.3.1 โดยทำการประยุกต์ใช้แบบจำลองทดแทนมากกว่าหนึ่งชนิด (ขึ้นอยู่กับประสบการณ์ผู้ออกแบบ)
- 4: ทำการสร้างประชากรลูก และหาค่าฟังก์ชันจริงของประชากรเหล่านั้น
- 5: หาผลเฉลยที่ไม่ถูกรอบจำกัดของยูเนียนเซตของประชากรลูกในข้อ 3 และ 4 และสมาชิกในเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอบจำกัดในรอบของการคำนวณก่อนหน้านี้
- 6: ถ้าขนาดของเมตริกซ์เก็บผลเฉลยใหญ่เกินไป คัดผลเฉลยบางตัวออกจากเมตริกซ์
- 7: กลับไปขั้นตอนที่ 2
- 8: หยุดการค้นหาผลเฉลย

นอกจากนี้ ยังมีการเชื่อมต่อระหว่างแบบจำลองทดแทนกับขั้นตอนวิธีวิวัฒนาการในรูปแบบอื่น เช่น การนำเอาวิธีการหาค่าเหมาะที่สุดแบบใช้อนุพันธ์มาช่วยในการหาผลเฉลยเหมาะสมที่สุดในเฟสของการหาผลเฉลยโดยใช้แบบจำลองทดแทน เป็นต้น

การเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการ ด้วยค่าประมาณของเกรเดียนต์และวิธีซิมเพล็กซ์ลงเขา

4.1 ความนำ

ในบทนี้ เป็นการแสดงการผสมผสานกระบวนการหาผลเฉลยเหมาะที่สุดด้วยขั้นตอนวิธีวิวัฒนาการ กับแนวคิดจากวิธีการหาค่าเหมาะที่สุดวิธีอื่น ในที่นี้คือการประมาณค่าเกรเดียนต์และวิธีซิมเพล็กซ์ลงเขา เนื้อหาในบทนี้เริ่มต้นด้วยขั้นตอนวิธีวิวัฒนาการชนิดใหม่ที่ได้จากการวิจัยนี้ นั่นคือวิธี PBIL แบบใช้จำนวนจริงเป็นตัวแปรออกแบบซึ่งเขียนแทนด้วย RPBIL ย่อมาจาก real-code population-based incremental learning จากนั้นจะนำเสนอการผสมผสานระหว่าง PBIL กับการประมาณค่าเกรเดียนต์ APG และทดสอบสมรรถนะของขั้นตอนวิธีการใหม่นี้ ในส่วนสุดท้ายเป็นการกล่าวถึงการเชื่อมต่อขั้นตอนวิธีวิวัฒนาการกับวิธีซิมเพล็กซ์ลงเขา ซึ่งเป็นกระบวนการหาค่าเหมาะที่สุดที่มีสมรรถนะสูงวิธีหนึ่ง

4.2 วิธี RPBIL

ในเบื้องต้น วิธี PBIL ถูกพัฒนาขึ้นมาเป็นวิธีหาค่าเหมาะที่สุดที่ใช้ตัวเลขฐานสอง (binary code) แทนตัวแปรออกแบบในการหาผลเฉลยเหมาะที่สุด ในหัวข้อนี้ อาจกล่าวได้ว่า เป็นการขยายผลต่อจาก PBIL เดิมเพื่อสามารถดำเนินการได้ด้วยเวกเตอร์ออกแบบจำนวนจริง (real code) โดยทั่วไป ขั้นตอนวิธีวิวัฒนาการหรือวิธีศึกษาสำนึกขั้นสูง ใช้ในการหาผลเฉลยเหมาะที่สุดที่มีเงื่อนไขบังคับของตัวแปรออกแบบ (bound-constrained optimisation) ซึ่ง ในกรณีของปัญหาการออกแบบที่มีฟังก์ชันเป้าหมายเดียวสามารถแสดงได้ดังนี้

$$\begin{aligned} &\text{Min } f(\mathbf{x}) \\ &L_i \leq x_i \leq U_i; i = 1, \dots, n \\ &\mathbf{x} \in R^n \end{aligned} \tag{4.1}$$

เมื่อ $\mathbf{x}$ เวกเตอร์ออกแบบขนาด $n \times 1$, f คือฟังก์ชันเป้าหมาย, L_i คือขอบเขตล่างของ $\mathbf{x}$, และ U_i คือขอบเขตบนของ $\mathbf{x}$

หลักการพื้นฐานของวิธี PBIL คือ การใช้เวกเตอร์หรือเมตริกซ์ของความน่าจะเป็น เป็นตัวแทนของประชากรในแต่ละรอบของการคำนวณ การวิวัฒนาการจะไม่เกิดที่ประชากรโดยตรง แต่จะกระทำที่เวกเตอร์หรือเมตริกซ์ตัวแทนนั้น ในกรณีของ RPBIL จะใช้เมตริกซ์ของความน่าจะเป็นเพื่อเป็นตัวแทนของประชากรของเวกเตอร์ออกแบบ ซึ่งในที่นี้เขียนแทนด้วย $\mathbf{P}$ หรือ P_{ij} ขนาด $n \times m$ เมื่อ m คือจำนวนคอลัมน์ ซึ่งแสดงถึงการแบ่งช่วงของขอบเขตของตัวแปรออกแบบ $[L_i, U_i]$ ออกเป็น m ช่วงย่อย (subintervals)

สมาชิกใดๆของเมตริกซ์ P_{ij} แสดงถึงค่าความน่าจะเป็นในการกำเนิดตัวแปร x_i ในช่วงของ $[L_i + (j-1)\delta_i, L_i + j\delta_i]$ เมื่อ $\delta_i = (U_i - L_i)/m$

สำหรับกระบวนการหาผลเฉลยเหมาะที่สุดของวิธี RPBIL เริ่มต้นด้วยเมตริกซ์ของความน่าจะเป็น $\mathbf{P}$ ที่มีค่าของแต่ละเอลิเมนต์เป็น $P_{ij} = 1/m$ จากนั้น ทำการสร้างประชากรจากเมตริกซ์นี้ และทำการหา $\mathbf{x}^{best}$ ผลเฉลยที่ให้ค่าฟังก์ชันเป้าหมายเหมาะที่สุด เมื่อทราบผลเฉลยที่ดีที่สุดในรอบของการคำนวณนั้น ทำการปรับปรุงค่าของ P_{ij} เทียบกับผลเฉลย $\mathbf{x}^{best}$ ดังนี้

$$P'_{ij} = (1 - L_{R,j})P_{ij}^{old} + L_{R,j} \quad (4.2)$$

เมื่อ

$$L_{R,j} = 0.5 \exp(-(j - r)^2). \quad (4.3)$$

เลขดัชนี r คือตำแหน่งที่เอลิเมนต์ x_i^{best} อยู่ในช่วงย่อย $[L_i + (r-1)\delta_i, L_i + r\delta_i]$ ค่าอัตราการเรียนรู้ (learning rate) L_R ในสมการ (4.3) ถูกกำหนดเพื่อป้องกันการลู่เข้าก่อนเวลาอันควรของ RPBIL เพื่อรักษาเงื่อนไข $\sum_{j=1}^T P_{ij} = 1$ แถวที่ i ของ $\mathbf{P}'$ จะถูกดัดแปลงเป็น

$$P''_{ij} = \frac{1}{\sum_{j=1}^m P'_{ij}} P'_{ij} \quad (4.4)$$

กระบวนการปรับปรุงค่าของ P_{ij} เทียบกับเวกเตอร์ออกแบบ $\mathbf{x}^{best}$ จะได้เมตริกซ์ $\mathbf{P}$ ในขั้นตอนสุดท้ายจากสมการ (4.4) ทำการปรับปรุงค่าของ $\mathbf{P}$ และ $\mathbf{x}^{best}$ แบบวนซ้ำ จนกระทั่งครบเงื่อนไขการหยุดค้นหาผลเฉลย กระบวนการคำนวณของ RPBIL มีดังต่อไปนี้ กำหนดให้ t คือหมายเลขวงวน, N_G จำนวนวงวนสูงสุด และ N_P คือขนาดของประชากร

ตัวแปรขาเข้า: N_G, N_P, n, m, L_R

ตัวแปรขาออก: $\mathbf{x}^{best}, f^{best}$

กระบวนการเริ่มต้น: $P_{ij} = 1/m, \delta_i = (U_i - L_i)/m, \mathbf{x}^{best}(0) = \{\}$

1: สำหรับ $t = 1$ ถึง N_G

2: สร้างเซตของเวกเตอร์ออกแบบ $\mathbf{X}(t)$ จาก P_{ij}

2.1: สำหรับ $i = 1$ ถึง n

2.2: สำหรับ $j = 1$ ถึง m

2.3: สุ่มสร้างเอลิเมนต์ x_i จำนวน $N_P \cdot P_{ij}$ ค่า ในช่วงย่อย $[L_i + (j-1)\delta_i, L_i + j\delta_i]$.

2.4: วงวนของดัชนี j

2.5: สลับตำแหน่งของ x_i ทั้ง N_P ตัวจาก 2.2-2.4

- 2.6: สมาชิกในแถวที่ i ของ $\mathbf{X}(t)$ คือ x_i จาก 2.5
- 2.7: วววนของดัชนี i
- 3: หาค่าฟังก์ชันเป้าหมาย $\mathbf{f}(t) = \text{fun}(\mathbf{X}(t))$.
- 4: หา $\mathbf{x}^{\text{best}}(t)$ จาก $\mathbf{X} \cup \mathbf{x}^{\text{best}}(t-1)$.
- 5: ปรับปรุงค่าของ P_{ij} เทียบกับ $\mathbf{x}^{\text{best}}$ ในวววนปัจจุบัน
 - 5.1: สำหรับ $i = 1$ ถึง n
 - 5.2: หาค่าดัชนี r โดยที่ $x_i^{\text{best}} \in [L_i + (r-1)\delta_i, L_i + r\delta_i]$.
 - 5.3: ปรับปรุง P_{ij} ด้วยสมการ (4.2-4.4)
 - 5.4: วววนของดัชนี i
- 6: วววนของดัชนี t

4.3 วิธี RPBIL แบบผสมผสาน

4.3.1 ทิศทางวิวัฒนาการ

ตัวดำเนินการทิศทางวิวัฒนาการนำเสนอใน [16] ซึ่งมีรูปแบบคล้ายกับหลักการหาผลเฉลยของวิธี PSO กระบวนการคำนวณทิศทางวิวัฒนาการหาได้จากการเลือก สมาชิกของประชากร ณ. วววนปัจจุบันของ RPBIL แบบสุ่มขึ้นมา 3 เวกเตอร์ $\mathbf{x}_1$, $\mathbf{x}_2$, และ $\mathbf{x}_3$ กำหนดให้ $\mathbf{x}_1$ เป็นตัวที่ดีที่สุด ทิศทางวิวัฒนาการหาได้จาก

$$\mathbf{s} = (\mathbf{x}_1 - \mathbf{x}_2) + (\mathbf{x}_1 - \mathbf{x}_3) + \mathbf{c} \quad (4.5)$$

เมื่อ $c_i = \varepsilon \cdot \text{randn}$, และ randn คือตัวเลขสุ่มที่กระจายแบบปกติด้วยค่าเฉลี่ยศูนย์และส่วนเบี่ยงเบนมาตรฐานเป็นหนึ่ง ค่าพารามิเตอร์ ε กำหนดให้เป็น 0.05 จากผลเฉลยทั้งสามที่เลือกมา ประชากรถูกชุดใหม่สองเวกเตอร์ที่ได้จากการดำเนินการนี้คือ

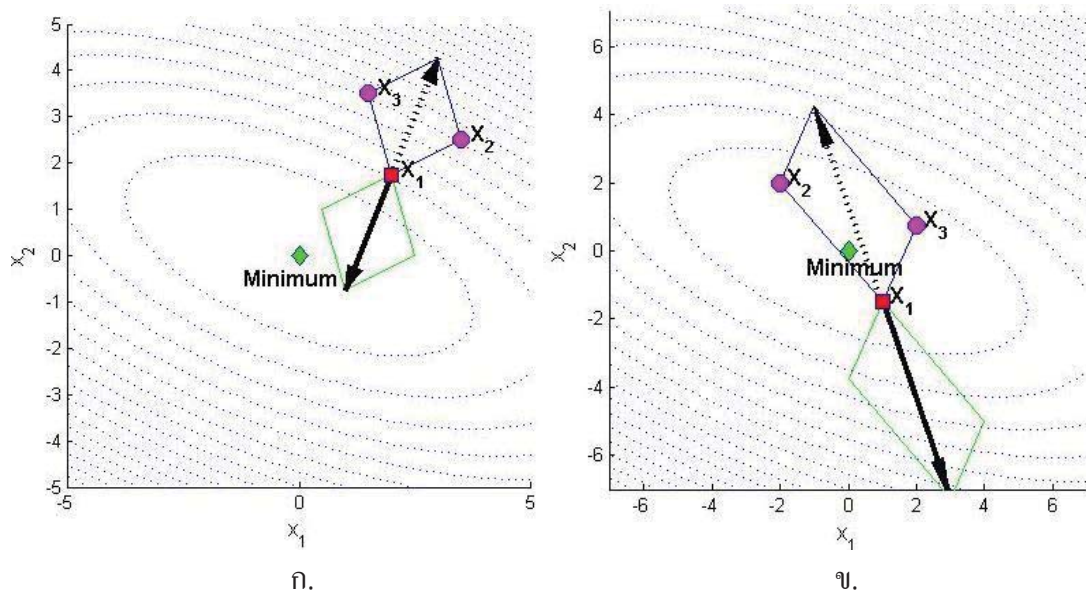
$$\begin{aligned} \mathbf{y}_1 &= \mathbf{x}_1 + \text{rand} \cdot \lambda(t) \mathbf{s} \\ \mathbf{y}_2 &= \mathbf{x}_1 - \text{rand} \cdot \lambda(t) \mathbf{s} \end{aligned} \quad (4.6)$$

เมื่อ $\text{rand} \in [0,1]$ คือตัวเลขสุ่มเอกกรูป (uniform random number) และ $\lambda(t)$ คือค่าความยาวของการก้าว (step length) ที่วววนที่ t ซึ่งสามารถคำนวณได้จากความสัมพันธ์

$$\lambda(t) = \exp\left(\frac{6.2146}{N_G - 1} - 0.6931\right) \exp\left(-\frac{6.2146}{N_G - 1} t\right), \quad (4.7)$$

โดยกำหนดให้ $\lambda(1) = 0.5$ และ $\lambda(N_G) = 0.001$ ค่าของ λ จะมีค่าสูงในรอบของการคำนวณเริ่มต้น เมื่อกระบวนการค้นหาผลเฉลยดำเนินไป λ จะมีค่าลดลงเพราะต้องการ การเปลี่ยนแปลงเล็กน้อยจากการดำเนินการนี้ เมื่อประชากรของเวกเตอร์ออกแบบเข้าใกล้ผลเฉลยเหมาะสมที่สุด รูปที่ 4-1 แสดงการดำเนินการ

แบบทิศทางวิวัฒนาการ จากรูปจะเห็นได้ว่าสมการทั้งสอง (4.6) ถูกตั้งไว้เพื่อเหตุการณ์ที่เป็นไปได้ทั้งสองกรณี ดังรูปที่ 4-1 ก. และ ข.



รูปที่ 4-1 การดำเนินการด้วยทิศทางวิวัฒนาการ

4.3.2 การประมาณค่าเกรเดียนต์

การประมาณค่าเกรเดียนต์ในที่นี้คือ ตัวดำเนินการทางวิวัฒนาการที่ใช้ทฤษฎีของอนุพันธ์เชิงทิศทาง (directional derivative) ในการประมาณค่าเกรเดียนต์ที่ตำแหน่งเวกเตอร์ออกแบบใดๆ จากข้อมูลของประชากรของผลเฉลยและค่าฟังก์ชันของผลเฉลยเหล่านั้น สำหรับฟังก์ชันใน n มิติใดๆ อนุพันธ์ของฟังก์ชัน f ในทิศทางของเวกเตอร์ s ที่จุด x เขียนแทนด้วย df/ds สามารถคำนวณได้จาก

$$u_s^T \nabla f = \frac{df}{ds} \approx \frac{f(x) - f(x + \Delta x)}{\Delta s} \quad (4.8)$$

เมื่อ u_s คือเวกเตอร์หน่วยของ s จากนิยามข้างบน กำหนดให้ประชากรในรอบปัจจุบันคือเมตริกซ์ขนาด $n \times N_G$ $\{x_1, \dots, x_{NG}\}$ และค่าฟังก์ชันเป้าหมายของผลเฉลยเหล่านั้นคือ $\{f_1, \dots, f_{NG}\}$ ผลเฉลยที่ดีที่สุดในวงวนปัจจุบันคือ x^{best} ซึ่งมีค่าฟังก์ชันเป้าหมายเป็น f^{best} เวกเตอร์เกรเดียนต์โดยประมาณของ f ที่จุด x^{best} สามารถคำนวณได้จากความสัมพันธ์

$$\begin{bmatrix} (x^{best} - x_1)^T \\ \vdots \\ (x^{best} - x_R)^T \end{bmatrix} \nabla f = A \nabla f = \begin{bmatrix} f^{best} - f_1 \\ \vdots \\ f^{best} - f_R \end{bmatrix} = b. \quad (4.9)$$

ในทางปฏิบัติ ใช้ผลเฉลย R ค่าจากเมตริกซ์ประชากร โดยที่เวกเตอร์เหล่านี้มีระยะใกล้กับ x^{best} มากที่สุด ค่าของ R ควรจะมีค่ามากกว่า n เพื่อความแม่นยำในการประมาณค่าเกรเดียนต์ ด้วยเหตุนี้ระบบสมการ

เชิงเส้น (4.9) เป็นระบบที่มีตัวแปร n ตัวและมี R สมการ ในการหาผลเฉลยของระบบสมการเชิงเส้นนี้สามารถทำได้โดยใช้ตัวดำเนินการเมตริกซ์ผกผันเทียม (pseudo-inverse) หรืออาจใช้วิธีกำลังสองต่ำสุด

ในการหาค่าเหมาะที่สุดด้วยวิธี steepest decent method เวกเตอร์ทิศทางในการหาค่าตอบคือ เวกเตอร์ที่มีทิศทางตรงข้ามกับ ∇f ดังนั้นเกรเดียนต์โดยประมาณที่ได้จากสมการ (4.9) สามารถนำมาใช้เป็นทิศทางในการหาผลเฉลยในรูปแบบคล้ายกัน กล่าวคือ $\mathbf{s} = -\nabla f$ เพื่อให้การประมาณค่าเกรเดียนต์นี้มีประโยชน์มากขึ้น ทำการประยุกต์การประมาณในช่วงด้วยฟังก์ชันกำลังสองหนึ่งครั้งในการหาผลเฉลยด้วยเกรเดียนต์โดยประมาณ ทำการหาเวกเตอร์ออกแบบในแนวทิศทาง $\mathbf{s}$ เพิ่ม ดังนี้

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{x}^{best} + \beta_1 \cdot \mathbf{s} \\ \mathbf{z}_2 &= \mathbf{x}^{best} + \beta_2 \cdot \mathbf{s}. \end{aligned} \quad (4.10)$$

เมื่อ $\beta_1 \neq \beta_2$ คือพารามิเตอร์ที่สุ่มค่าขึ้นมาในช่วงของ $(0,1]$ ขั้นตอนถัดไป ประยุกต์ใช้การประมาณค่าในช่วงเพื่อหาค่า β ที่ให้ค่าฟังก์ชันต่ำสุด ในกรณีของฟังก์ชันกำลังสอง

$$f(\mathbf{x}^{best} + \beta \cdot \mathbf{s}) = C_1 \beta^2 + C_2 \beta + C_3 \quad (4.11)$$

จะได้เมตริกซ์สัมประสิทธิ์ของฟังก์ชันกำลังสองจากการแก้ระบบสมการเชิงเส้น

$$\begin{bmatrix} 0 & 0 & 1 \\ \beta_1^2 & \beta_1 & 1 \\ \beta_2^2 & \beta_2 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} f^{best} \\ f(\mathbf{z}_1) \\ f(\mathbf{z}_2) \end{bmatrix} \quad (4.12)$$

เมื่อ $f(\mathbf{z}_1)$ และ $f(\mathbf{z}_2)$ คือค่าฟังก์ชันเป้าหมายที่จุด $\mathbf{z}_1$ และ $\mathbf{z}_2$ ตามลำดับ จากนั้น เวกเตอร์ตัวแปรออกแบบตัวที่สามหาได้จาก

$$\mathbf{z}_3 = \mathbf{x}^{best} + \beta_3 \cdot \mathbf{s}. \quad (4.13)$$

เมื่อ

$$\beta_3 = -C_2/2/C_1.$$

รูปที่ 4-2 แสดงกระบวนการหาผลเฉลยตามแนวของทิศทางตรงข้ามกับเวกเตอร์เกรเดียนต์ โดยการกำเนิดเวกเตอร์ออกแบบสองจุดแรก $\mathbf{z}_1$ และ $\mathbf{z}_2$ จากนั้น ทำการหาจุด $\mathbf{z}_3$ ด้วยสมการ (4.13)

4.3.3 ขั้นตอนวิธีแบบผสม

การผสมผสาน RPBIL กับตัวดำเนินการทิศทางวิวัฒนาการและเกรเดียนต์โดยประมาณ สามารถทำได้โดยให้ RPBIL เป็นกระบวนการออกแบบหลัก ส่วนตัวดำเนินการทั้งสองจะเป็นตัวช่วยในการผลิตประชากรลูกของ RPBIL ในแต่ละรอบของการคำนวณ กระบวนการคำนวณแบบผสมดังกล่าว มีรายละเอียดดังนี้

ตัวแปรขาเข้า: $N_G, N_P, n, m, L_R, \varepsilon$

ตัวแปรขาออก: $\mathbf{x}^{\text{best}}, f^{\text{best}}$

ขั้นตอนเริ่มต้น: $P_{ij} = 1/m, \delta_i = (U_i - L_i)/m, \mathbf{X}(0), \mathbf{f}(0), \mathbf{x}^{\text{best}}(0)$

1: สำหรับ $t = 1$ ถึง N_G

2: สร้างประชากรย่อย $\mathbf{X}_1$ จำนวน $N_P/2$ จากเมตริกซ์ของความน่าจะเป็น P_{ij}

3: คำนวณเกรเดียนต์โดยประมาณ จากสมการ (4.9) ซึ่งจะได้เวกเตอร์ทิศทาง $\mathbf{s}$

4: หา $\mathbf{X}_2$ จากตัวดำเนินการเกรเดียนต์โดยประมาณซึ่งมี 3 เวกเตอร์ดังในรูปที่ 4-2

5: สร้างเวกเตอร์ตัวแปรออกแบบจำนวน $N_P/2 - 3$ เวกเตอร์เขียนแทนด้วย $\mathbf{X}_3$ โดยใช้ตัวดำเนินการ

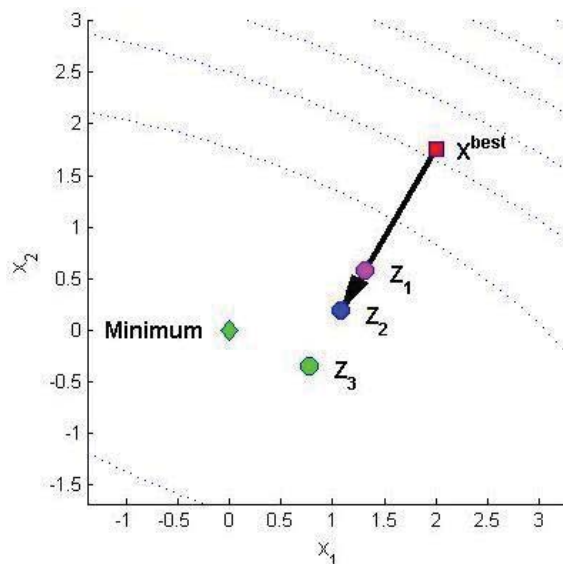
ทิศทางวิวัฒนาการ

6: รวมประชากรย่อยทั้งสาม $\mathbf{X}(t) = \mathbf{X}_1 \cup \mathbf{X}_2 \cup \mathbf{X}_3$ และหาค่าฟังก์ชันเป้าหมาย $\mathbf{f}(t) = \text{fun}(\mathbf{X}(t))$.

7: หา $\mathbf{x}^{\text{best}}(t)$ จาก $\mathbf{X} \cup \mathbf{x}^{\text{best}}(t-1)$.

8: ปรับปรุงค่าของ P_{ij} เทียบกับ $\mathbf{x}^{\text{best}}$

9: วนวนของดัชนี t



รูปที่ 4-2 การหาความยาวในการก้าวของเวกเตอร์เกรเดียนต์แบบประมาณ

4.4 การทดสอบสมรรถนะของขั้นตอนวิธีวิวัฒนาการใหม่

4.4.1 ฟังก์ชันทดสอบและวิธีวิวัฒนาการอื่น

เพื่อทำการทดสอบสมรรถนะในการหาผลเฉลยของขั้นตอนวิธีวิวัฒนาการแบบผสมที่พัฒนาขึ้นมาใหม่ ได้ดำเนินการประยุกต์ใช้วิธีหาค่าเหมาะที่สุดชนิดใหม่นี้ในการหาผลเฉลยเหมาะที่สุดที่มีเงื่อนไขบังคับของตัวแปรออกแบบซึ่งเรียกว่าฟังก์ชันทดสอบซึ่งมีรายละเอียดดังแสดงในตารางที่ 4-1 ซึ่งประกอบด้วยฟังก์ชันเป้าหมายที่มีผลเฉลยเหมาะที่สุดค่าเดียวและหลายค่า ซึ่งกรณีหลังเป็นการหาค่าเหมาะที่สุดของทั้งหมด นอกจากการประยุกต์ใช้วิธีวิวัฒนาการแบบผสมนี้แล้ว ยังทำการประยุกต์ใช้วิธีวิวัฒนาการที่พัฒนามาก่อนเพื่อเปรียบเทียบกับวิธีใหม่ดังนี้

- Real-code ant colony optimization (ACO)
- Charged system search (CSS)
- Differential evolution (DE)
- Continuous tabu search (TS)
- Fireworks algorithm (FA)
- Binary-code genetic algorithm (GA)
- Particle swarm optimization
- Simulated annealing (SA)
- Continuous scatter search (SS)
- Binary PBIL (PB)

วิธีวิวัฒนาการใหม่ที่น่าสนใจในบทนี้มีสองวิธีกล่าวคือ วิธี RPBIL ดังรายละเอียดในหัวข้อย่อยที่ 4.2 ในที่นี้จะเขียนแทนด้วย PR วิธีที่สองคือวิธี RPBIL แบบผสมดังรายละเอียดในหัวข้อ 4.3 ในที่นี้จะเขียนแทนด้วย HPR ขั้นตอนวิธีวิวัฒนาการทั้งหมดถูกใช้ในการหาผลเฉลยของฟังก์ชันทดสอบในตารางที่ 4-1 โดยใช้จำนวนวนทั้งหมดเป็น $10n$ และขนาดของประชากรคือ $7n$ แต่ละวิธีจะถูกประยุกต์ใช้หาผลเฉลยของปัญหาการออกแบบ 30 ครั้ง ทั้งนี้เป็นเพราะว่าขั้นตอนวิธีวิวัฒนาการโดยทั่วไปไม่มีความสม่ำเสมอในการหาผลเฉลยต่ำมาก ดังนั้น ในการศึกษาการทดสอบสมรรถนะในการหาคำตอบที่เหมาะสมที่สุดของวิธีการเหล่านี้ จำเป็นต้องรันเพื่อหาคำตอบหลายๆครั้ง จึงจะสามารถเปรียบเทียบสมรรถนะของขั้นตอนวิธีวิวัฒนาการได้ถูกต้อง

ตารางที่ 4-1 ฟังก์ชันทดสอบ

ชื่อฟังก์ชัน และที่มา	$[L,U]^n$	ชื่อฟังก์ชัน และที่มา	$[L,U]^n$
1, B_2 [78]	$[-50,100]^2$	18, Griewangk [78]	$[-5.12,5.12]^{10}$
2, Beale [39]	$[-4.5,4.5]^2$	19, $Perm$ [39]	$[-15,15]^{15}$
3, Booth [39]	$[-10,10]^2$	20, $Perm0$ [39]	$[-15,15]^{15}$
4, Easom [78]	$[-100,100]^2$	21, Cigar [78]	$[-3,7]^{20}$
5, Goldstein & Price [78]	$[-2,2]^2$	22, Diagonal plane [78]	$[0.5,1.5]^{20}$
6, Martin & Gaddy [78]	$[-20,20]^2$	23, Dixon & Price [39]	$[-10,10]^{20}$
7, Matyas [39]	$[-5,10]^2$	24, Levy(n) [39]	$[-10,10]^{20}$
8, Penny & Linfield* [14]	$[-5,5]^2$	25, Powell(n) [39]	$[-4,5]^{20}$
9, Powersum [39]	$[0,2]^2$	26, Rastrigin(n) [78]	$[-2.56,5.12]^{20}$
10, Branin [78]	$[-5,15]^2$	27, Rosenbrock [78]	$[-5,5]^{20}$
11, Shubert [39]	$[-10,10]^2$	28, Sum Squares [39]	$[-5,10]^{20}$
12, Six Hump Camel Back [39]	$[-5,10]^2$	29, Schwefel [39]	$[-500,500]^{20}$
13, Colville [39]	$[-10,10]^4$	30, Trid(n) [39]	$[-400,400]^{20}$
14, Hartmann 3,4 [78]	$[0,4]^4$	31, Ackley(n) [39]	$[-15,30]^{30}$
15, Shekel [78]	$[0,10]^4$	32, Ellipsoid [78]	$[-3,7]^{30}$
16, Zakharov [78]	$[-5,10]^5$	33, Plane [78]	$[-0.5,1.5]^{30}$
17, Hartmann 6,4 [78]	$[0,6]^6$	34, Sphere [78]	$[-3,7]^{30}$
		35, Tablet [78]	$[-3,7]^{30}$

* $f_8(\mathbf{x}) = 0.5(x_1^4 - 16x_1^2 + 5x_1) + 0.5(x_2^4 - 16x_2^2 + 5x_2)$

4.4.2 ผลการทดสอบ

วิธีวิวัฒนาการทุกวิธีถูกใช้ในการหาผลเฉลยเหมาะสมที่สุดเป็นจำนวน 30 ครั้งในแต่ละปัญหาการหาค่าเหมาะที่สุด ค่าเฉลี่ยของฟังก์ชันเป้าหมายของผลเฉลยเหมาะสมที่สุดทั้ง 30 ค่าที่ได้จากการใช้วิธีที่เหมาะสมที่สุดเชิงวิวัฒนาการหาคำตอบ จะถูกแปลงค่าให้อยู่ในช่วง $[0, 1]$ เพื่อง่ายต่อการเปรียบเทียบดังนี้

$$\bar{f}_i = \left| \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right| \quad (4.14)$$

เมื่อ $f_{\min}$ ค่าฟังก์ชันเป้าหมายที่ดีที่สุดจากค่าฟังก์ชันเป้าหมายทั้งหมดที่มีอยู่ และ $f_{\max}$ ค่าฟังก์ชันเป้าหมายสูงสุดที่ได้ สำหรับปัญหาการออกแบบใดๆ จากเงื่อนไขที่กล่าวมา วิธีที่ดีที่สุดของแต่ละปัญหาการออกแบบจะมีค่าฟังก์ชันเป้าหมายเฉลี่ยเป็น $\bar{f} = 0$ ส่วนวิธีที่แย่ที่สุดจะให้ค่าฟังก์ชันเป้าหมายเฉลี่ยคือ $\bar{f} = 1$

ผลจากการเปรียบเทียบแสดงในตารางที่ 4-2 วิธีที่ดีที่สุดเรียงตามลำดับคือวิธี HPR ซึ่งเป็นวิธีผสมที่นำเสนอในที่นี้ นอกจากนั้นก็มีวิธี ACO, TS, FA, PR, และ SA ตามลำดับ

ตารางที่ 4-2 การเปรียบเทียบค่าฟังก์ชันเป้าหมายเฉลี่ย

Fn No.	ACO	CSS	DE	TS	FA	GA	PSO	SA	SS	PB	PR	HPR
1	0.000	0.035	0.040	0.016	0.050	0.869	0.018	0.032	0.105	1.000	0.019	0.005
2	0.020	0.520	1.000	0.130	0.220	0.265	0.000	0.148	0.542	0.302	0.431	0.058
3	0.014	0.056	0.166	0.005	0.056	1.000	0.015	0.353	0.413	0.778	0.014	0.000
4	0.829	1.000	0.830	0.553	0.997	0.550	0.283	0.988	1.000	0.912	0.529	0.000
5	0.015	1.000	0.439	0.000	0.090	0.836	0.520	0.718	0.687	0.841	0.024	0.144
6	0.001	0.051	0.068	0.003	0.037	0.400	0.001	0.083	0.089	1.000	0.005	0.000
7	0.036	0.109	0.293	0.006	0.099	0.500	0.006	0.501	0.578	1.000	0.016	0.001
8	0.085	0.074	0.051	0.516	0.284	0.159	1.000	0.000	0.193	0.185	0.190	0.188
9	0.009	0.030	0.132	0.002	0.043	0.440	0.119	1.000	0.153	0.658	0.002	0.002
10	1.000	0.076	0.516	0.016	0.149	0.421	0.163	0.069	0.290	0.755	0.041	0.000
11	0.934	0.520	0.862	0.165	0.315	0.346	0.539	0.116	1.000	0.570	0.340	0.000
12	0.038	0.801	0.476	0.273	0.363	0.536	0.998	0.337	0.749	1.000	0.125	0.000
13	0.239	0.186	1.000	0.031	0.319	0.329	0.141	0.154	0.337	0.344	0.165	0.000
14	0.000	1.000	0.002	0.005	0.006	0.036	0.301	0.439	0.023	0.203	0.084	0.004
15	0.052	0.000	0.821	0.623	0.885	0.634	0.454	0.920	0.377	1.000	0.772	0.573
16	0.033	0.017	0.412	0.001	0.020	0.632	0.339	0.180	0.205	1.000	0.006	0.000
17	0.000	1.000	0.051	0.237	0.129	0.191	0.908	0.001	0.531	0.405	0.251	0.247
18	0.704	0.000	1.000	0.163	0.175	0.076	0.150	0.161	0.015	0.040	0.173	0.097
19	0.000	0.462	0.000	0.546	0.619	0.027	0.890	0.000	0.267	0.081	1.000	0.417
20	0.000	0.000	0.002	0.000	0.000	0.000	0.000	1.000	0.000	0.001	0.000	0.000
21	0.000	0.000	0.036	0.002	0.026	0.338	0.179	0.001	0.199	1.000	0.043	0.000
22	0.000	0.014	0.000	0.006	0.000	0.001	1.000	0.000	0.057	0.022	0.022	0.000
23	0.009	0.000	0.114	0.000	0.013	0.166	0.308	0.003	0.022	1.000	0.042	0.007
24	0.030	0.023	0.088	1.000	0.886	0.059	0.320	0.000	0.042	0.212	0.403	0.408
25	0.047	0.000	0.147	0.006	0.011	0.420	0.712	0.005	0.049	1.000	0.061	0.001
26	1.000	0.386	0.588	0.601	0.485	0.176	0.477	0.000	0.470	0.293	0.722	0.379
27	0.076	0.011	0.759	0.000	0.081	0.481	1.000	0.004	0.117	0.860	0.153	0.087
28	0.001	0.000	0.042	0.002	0.024	0.149	0.810	0.001	0.157	1.000	0.036	0.000
29	1.000	0.266	0.468	0.449	0.485	0.130	0.827	0.000	0.795	0.223	0.679	0.507
30	0.110	0.333	0.080	0.429	0.000	0.410	1.000	0.454	0.698	0.515	0.332	0.203
31	0.279	0.074	0.519	0.323	0.264	0.537	1.000	0.000	0.642	0.889	0.331	0.079
32	0.004	0.000	0.041	0.001	0.036	0.119	1.000	0.000	0.179	0.452	0.031	0.004
33	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
34	0.007	0.000	0.065	0.000	0.009	0.106	1.000	0.000	0.187	0.319	0.016	0.000

35	0.000	0.000	0.004	0.008	0.001	0.030	1.000	0.000	0.025	0.054	0.232	0.002
total	6.573	8.043	11.113	6.118	7.177	11.370	17.477	7.669	11.192	19.915	7.291	3.414

* PB = binary PBIL, PR = real code PBIL, HPR =Hybrid real-code PBIL

ตารางที่ 4-3 แสดงการเปรียบเทียบสมรรถนะในการหาผลเฉลยด้วยวิธีการทางสถิติของปัญหาการออกแบบที่ 1 เนื่องจากมีวิธีวิวัฒนาการที่ใช้งานทั้งหมด 12 วิธี ดังนั้นตารางนี้มีขนาด 12×12 ที่ตำแหน่งแถวที่ I และคอลัมน์ที่ J เป็นผลการเปรียบเทียบค่าฟังก์ชันเป้าหมายที่ได้จากการรันวิธี I เทียบกับวิธี J ค่าตัวเลขที่ตำแหน่งแถวที่ I และคอลัมน์ที่ J จะมีค่าเป็น 1 เมื่อวิธีที่ I ดีกว่าวิธีที่ J อย่างมีนัยสำคัญที่ระดับความเชื่อมั่น 95% หากไม่ใช่ตำแหน่งนั้นมีค่าเป็น 0 เมื่อรวมผลจากการเปรียบเทียบทุกวิธีแล้ว ขั้นตอนวิธีวิวัฒนาการที่ดีที่สุดสำหรับฟังก์ชันทดสอบที่ 1 คือ ACO ส่วนวิธีที่ดีอันดับสองคือวิธี HPR

เมื่อรวมผลการวิเคราะห์เปรียบเทียบทางสถิติ ของปัญหาการออกแบบทุกฟังก์ชันทดสอบแล้ว ผลการเปรียบเทียบแบบเรียงลำดับ (ranking) แสดงในตารางที่ 4-4 จากผลในตาราง วิธีที่ดีที่สุดในภาพรวมคือวิธีผสม HPR ที่แสดงรายละเอียดในหัวข้อย่อยที่ 4.3 นั่นเอง ส่วนอีก 4 วิธีที่ให้สมรรถนะดีรองลงมาคือ TS, ACO, TSS และ SA

จากตรงนี้ สามารถสรุปได้ว่าวิธี RPBIL ที่พัฒนาขึ้นมาใหม่ เป็นวิธีที่ให้คำตอบเหมาะสมสุดในระดับที่ยอมรับได้ เมื่อทำการผสมผสานวิธี RPBIL กับตัวดำเนินการทิศทางวิวัฒนาการและเกรเดียนต์โดยประมาณแล้ว สมรรถนะในการหาผลเฉลยเหมาะสมสุดของวิธีผสมสูงขึ้นอย่างมีนัยสำคัญ แนวคิดของการประยุกต์ใช้ตัวดำเนินการทั้งสองกับขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายสามารถทำได้ และเป็นส่วนหนึ่งของงานวิจัยในอนาคตของผู้วิจัย

ตารางที่ 4-3 การเปรียบเทียบทางสถิติ

EAs	ACO	CSS	DE	TS	FA	GA	PSO	SA	SS	PB	PR	HPR
ACO	0	0	0	0	0	0	0	0	0	0	0	0
CSS	1	0	0	0	0	0	0	0	0	0	0	1
DE	1	0	0	1	0	0	1	0	0	0	0	1
TS	1	0	0	0	0	0	0	0	0	0	0	1
FA	1	0	0	1	0	0	1	0	0	0	0	1
GA	1	1	1	1	1	1	0	1	1	1	0	1
PSO	1	0	0	0	0	0	0	0	0	0	0	0
SA	1	0	0	1	0	0	0	0	0	0	0	1
SS	1	0	0	1	0	0	1	0	0	0	1	1
PB	1	1	1	1	1	1	0	1	1	1	0	1
PR	1	0	0	0	0	0	0	0	0	0	0	1
HPR	1	0	0	0	0	0	0	0	0	0	0	0
Total	11	2	2	6	2	0	5	2	2	0	5	9
Ranking	1	6	6	3	6	11	4	6	6	11	4	2

ตารางที่ 4-4 ผลการเปรียบเทียบเชิงสถิติด้วยการจัดลำดับ

Function No.	ACO	CSS	DE	TS	FA	GA	PSO	SA	SS	PB	PR	HPR
1	1	6	6	3	6	11	4	6	6	11	4	2
2	2	9	9	4	6	6	1	5	9	6	9	3
3	3	6	8	2	6	11	3	8	10	11	3	1
4	6	6	6	4	6	4	1	6	6	6	3	1
5	1	8	6	1	4	8	7	8	8	8	1	4
6	2	6	6	3	6	11	3	6	6	12	5	1
7	3	6	8	2	6	9	3	9	9	9	3	1
8	3	3	2	11	5	5	12	1	5	5	5	5
9	4	4	7	2	4	10	7	11	7	11	2	1
10	11	3	9	2	6	9	7	3	8	11	3	1
11	10	5	10	3	4	5	5	2	10	5	5	1
12	2	7	7	5	5	7	7	4	7	7	3	1
13	7	4	12	1	7	7	3	4	7	7	4	1
14	1	12	2	3	3	6	10	10	6	9	8	3
15	1	1	8	5	10	5	4	10	3	12	8	5
16	6	4	9	2	4	11	9	7	7	12	3	1
17	1	12	3	5	4	5	11	1	10	9	5	5
18	11	1	12	7	7	4	7	6	2	3	7	4
19	3	7	1	9	10	4	11	1	6	5	12	7
20	4	1	4	2	4	4	4	12	4	4	4	2
21	3	1	7	5	6	11	9	4	9	12	8	2
22	4	8	1	7	4	6	12	1	11	9	9	1
23	4	1	9	1	5	10	11	3	7	11	8	5
24	3	2	6	11	11	4	8	1	4	7	8	8
25	6	1	9	3	5	10	11	3	6	11	6	2
26	12	5	9	9	6	2	6	1	6	3	11	4
27	4	2	10	1	5	9	10	2	5	10	7	7
28	3	1	8	5	6	9	11	3	9	11	7	2
29	12	3	5	5	5	2	10	1	10	3	9	5
30	2	5	2	7	1	8	12	8	11	8	5	4
31	4	2	8	6	4	8	12	1	10	11	6	2
32	4	1	7	3	7	9	12	2	10	11	6	4
33	1	11	1	1	1	1	1	1	1	1	12	1
34	5	1	8	4	5	9	12	3	10	11	7	2
35	3	1	6	6	4	8	12	2	8	10	11	5
total	152	156	231	150	188	248	268	156	253	292	217	104

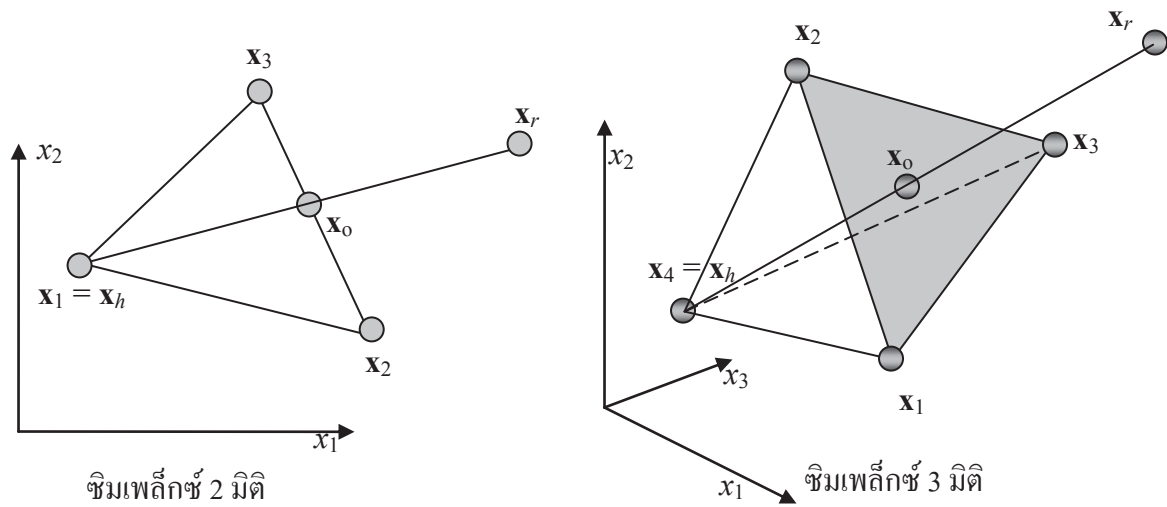
4.5 การผสมผสานขั้นตอนวิธีวิวัฒนาการกับวิธีซิมเพล็กซ์ลงเขา

4.5.1 วิธีซิมเพล็กซ์ลงเขา (Simplex Downhill Method, SPDH)

วิธีซิมเพล็กซ์ลงเขาเรียกอีกชื่อหนึ่งว่าวิธีเนลเดอร์และมิด (Nelder and Mead) ตามชื่อผู้พัฒนาวิธีการนี้ หรืออาจเรียกวิธีการค้นหาแบบรูปทรงหลายหน้ายืดหยุ่น (flexible polyhedron search) ซิมเพล็กซ์ในที่นี้หมายถึงรูปทรงเหลี่ยมที่เกิดจากจุดยอด $n + 1$ จุดในกรณีของปัญหาการออกแบบที่เหมาะสมที่สุดที่มีตัวแปรต้น n ตัว จากนั้น ซิมเพล็กซ์ในกรณีฟังก์ชันจุดประสงค์ 1 มิติคือเส้นตรงที่ลากเชื่อมระหว่างจุด 2 จุด ในกรณีของฟังก์ชัน 2 มิติ ซิมเพล็กซ์คือรูปสามเหลี่ยม และในกรณีของฟังก์ชัน 3 มิติ ซิมเพล็กซ์คือปริมาตรสามเหลี่ยม รูปที่ 4-3 แสดงซิมเพล็กซ์ใน 2 และ 3 มิติ

กระบวนการค้นหาผลเฉลยของวิธีหาค่าเหมาะที่สุดนี้สามารถจินตนาการว่าเป็นการเคลื่อนซิมเพล็กซ์ลงจากภูเขาในกรณีของการหาค่าต่ำสุด SPDH ใช้ตัวดำเนินการหลัก 3 ตัว กล่าวคือ การสะท้อน

(reflection), การขยาย (expansion) และ การหดตัว (contraction) รูปที่ 4-4 แสดงตัวดำเนินการแบบการสะท้อนซึ่งเป็นตัวดำเนินการหลักของ SPDH



รูปที่ 4-3 ซิมเพล็กซ์ใน 2 และ 3 มิติ และการสะท้อน (reflection)

กระบวนการค้นหาผลเฉลยเหมาะที่สุดของ SPDH เริ่มต้นด้วยซิมเพล็กซ์เริ่มต้น หรืออาจเรียกว่าเซตของผลเฉลยเริ่มต้นจำนวน $n + 1$ ตัว $S = \{ \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n+1} \}$ สำหรับการหาค่าเหมาะที่สุดใน n มิติ ตัวดำเนินการหลักของวิธีหาค่าเหมาะที่สุดนี้คือการสะท้อน ซึ่งดำเนินการดังนี้

หาผลเฉลย $\mathbf{x}_h$ และ $\mathbf{x}_l$ ซึ่งเป็นสมาชิกของ S โดยที่ $\mathbf{x}_h$ มีค่าฟังก์ชันจุดประสงค์สูงสุด และ $\mathbf{x}_l$ มีค่าฟังก์ชันจุดประสงค์ต่ำสุด กระบวนการสะท้อนหมายถึงการสร้างผลเฉลยใหม่ $\mathbf{x}_r$ จากสมการ

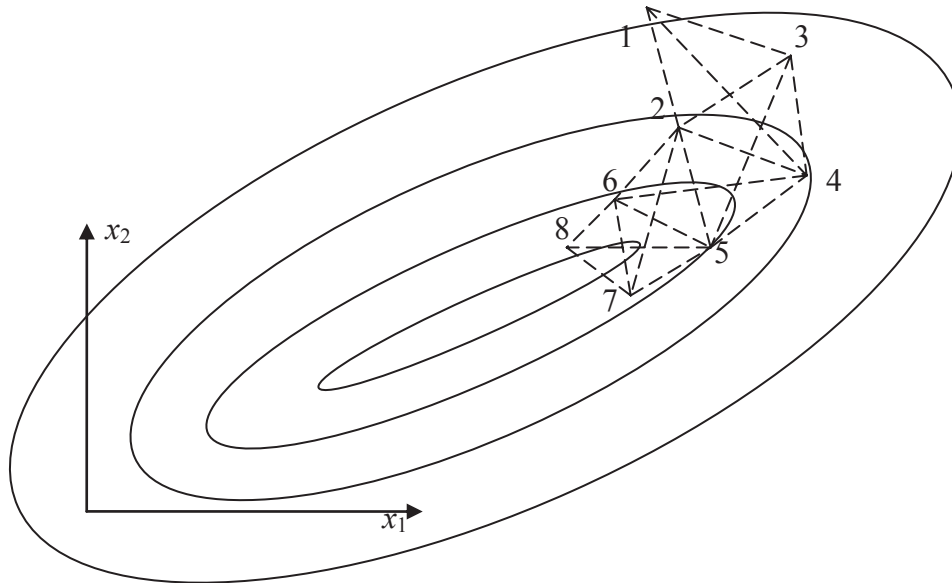
$$\mathbf{x}_r = (1+\alpha)\mathbf{x}_o - \alpha\mathbf{x}_h \quad (4.15)$$

เมื่อ

$$\mathbf{x}_o = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq h}}^{n+1} \mathbf{x}_i \quad (4.16)$$

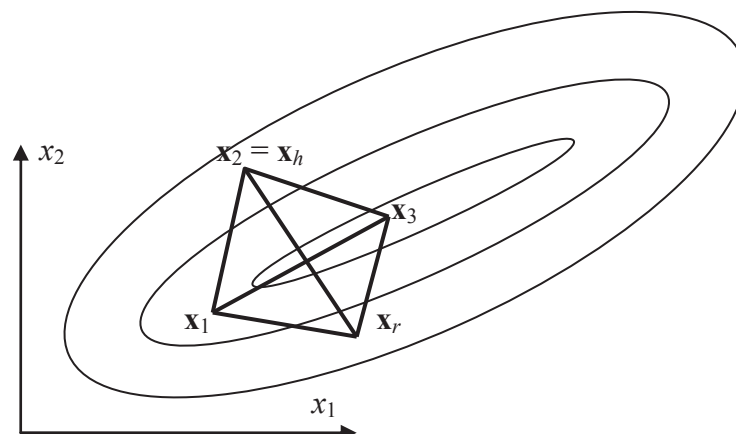
และ $\alpha > 0$ คือสัมประสิทธิ์การสะท้อน

ถ้า $f(\mathbf{x}_r)$ มีค่าน้อยกว่า $f(\mathbf{x}_h)$ แทนผลเฉลย $\mathbf{x}_h$ ด้วย $\mathbf{x}_r$ นั้นจะเกิดเป็นซิมเพล็กซ์อันใหม่ที่เคลื่อนที่ลงต่ำกว่าซิมเพล็กซ์อันแรก รูปที่ 4-5 แสดงการเคลื่อนที่ของซิมเพล็กซ์ลงไปยังตำแหน่งที่มีค่าฟังก์ชันจุดประสงค์ต่ำกว่าเดิม จากรูป ซิมเพล็กซ์เริ่มต้นเกิดจากจุดยอด 1-2-3 หลังจากนั้น ซิมเพล็กซ์เคลื่อนที่ลงต่ำไปตามตำแหน่งของจุดยอดของซิมเพล็กซ์ ดังนี้ 2-3-4, 2-4-5, 2-6-5, 5-6-7 และ 6-7-8



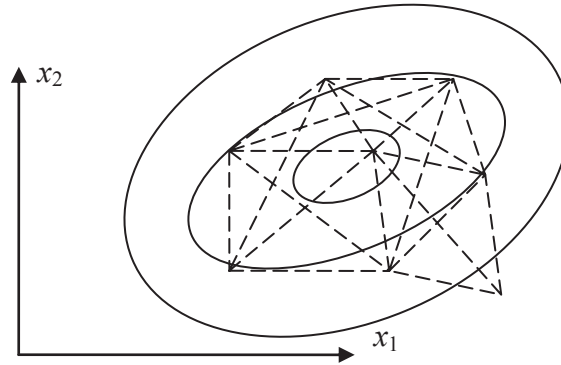
รูปที่ 4-4 การดำเนินการแบบการสะท้อนไปยังจุดที่เหมาะสมที่สุด

อย่างไรก็ตาม การประยุกต์ใช้ตัวดำเนินการสะท้อนอย่างเดียวไม่สามารถทำให้วิธีการหาค่าเหมาะที่สุดนี้ค้นหาผลเฉลยที่เหมาะสมที่สุดได้ รูปที่ 4-6 แสดงกรณีที่เป็นจุดอ่อนของวิธีซิมเพล็กซ์ลงเขา จากรูปซิมเพล็กซ์เริ่มต้นเกิดจากจุดยอด x_1 , x_2 และ x_3 ซึ่งพบว่า x_2 ให้ค่าฟังก์ชันจุดประสงค์สูงสุดจึงถูกตั้งเป็นจุด x_h เมื่อดำเนินการสะท้อนจะได้ผลเฉลยใหม่คือจุด x_r และซิมเพล็กซ์อันใหม่คือ x_1 , x_3 และ x_r จากรูป หากเราดำเนินการสะท้อนต่อไปจุดที่ให้ค่าฟังก์ชันจุดประสงค์สูงสุดคือจุด $x_h = x_r$ เมื่อดำเนินการแบบสะท้อน ผลเฉลยใหม่ที่ได้จะเป็นจุด x_2 เมื่อดำเนินการซ้ำไปเรื่อยๆ ซิมเพล็กซ์จะสลับไป-มา ระหว่างซิมเพล็กซ์ x_1 - x_2 - x_3 และ x_1 - x_3 - x_r ส่งผลให้กระบวนการค้นหาผลเฉลยไม่สามารถหาคำตอบได้ ปรากฏการณ์นี้เรียกว่า กรณีของซิมเพล็กซ์ร่อมหุบเขา วิธีแก้ปัญหาคือให้ซิมเพล็กซ์เคลื่อนที่ออกจากบริเวณนี้ได้คือ ในการดำเนินการแบบสะท้อนครั้งแรก จะไม่ตัด x_2 ทิ้ง แต่จะตัดตัวที่ให้ค่าฟังก์ชันจุดประสงค์สูงสุดอันดับสองซึ่งนั่นคือ x_1 ฉะนั้น ซิมเพล็กซ์ใหม่หลังจากการสะท้อนคือ x_2 - x_3 - x_r



รูปที่ 4-5 ซิมเพล็กซ์ร่อมหุบเขา

ปัญหาอีกรูปแบบหนึ่งที่มีโอกาสเกิดขึ้นได้ในขณะที่ดำเนินการแบบการสะท้อนคือการที่ซิมเพล็กซ์เคลื่อนที่วนไปรอบๆบริเวณใกล้เคียงกับจุดเหมาะที่สุด แต่ไม่สามารถเข้าถึงตำแหน่งของจุดเหมาะที่สุดได้ ดังแสดงในรูปที่ 4-6 ปัญหาเหล่านี้สามารถแก้ไขได้โดยการเพิ่มจำนวนของตัวดำเนินการเข้าไปในขั้นตอนวิธีซิมเพล็กซ์ลงเขา ตัวดำเนินการที่เพิ่มเข้ามาคือการขยายและการหดตัว



รูปที่ 4-6 ซิมเพล็กซ์วนซ้ำบริเวณเดิม

สำหรับตัวดำเนินการเพิ่มเติม ตัวดำเนินการแรกคือการขยายซึ่งจะถูกเรียกให้ทำงานหลังจากการสะท้อนเมื่อ $f(\mathbf{x}_r) < f(\mathbf{x}_l)$ จากเงื่อนไขดังกล่าว เราคาดว่าทิศทางการเคลื่อนที่นี้จะน่าจะช่วยลดค่าฟังก์ชันลงไปได้อีก ดังนั้น การขยายคือการก้าวไปข้างหน้าต่อจากจุด $\mathbf{x}_r$ ไปในทิศทางเดิมที่ใช้ในขั้นตอนการสะท้อน ผลเฉลยใหม่ที่ได้จากการขยายคือ $\mathbf{x}_e$ ซึ่งสามารถคำนวณได้ดังนี้

$$\mathbf{x}_e = (1-\gamma)\mathbf{x}_o + \gamma\mathbf{x}_r \quad (4.17)$$

เมื่อ $\gamma > 1$ เรียกว่าสัมประสิทธิ์การขยาย

ในกรณีที่ $f(\mathbf{x}_r) > f(\mathbf{x}_h)$ ตัวดำเนินการแบบการหดตัวจะถูกเรียกให้ดำเนินการ ส่งผลให้ได้ผลเฉลยใหม่ $\mathbf{x}_c$ ซึ่งสามารถคำนวณได้ดังนี้

$$\mathbf{x}_c = (1-\beta)\mathbf{x}_o + \beta\mathbf{x}_h \quad (4.18)$$

เมื่อ $0 \leq \beta \leq 1$ คือค่าสัมประสิทธิ์ของการหดตัว โดยมีความคาดหมายว่า $f(\mathbf{x}_c) < f(\mathbf{x}_h)$ อย่างไรก็ตาม ถ้า $f(\mathbf{x}_c) \geq f(\mathbf{x}_h)$ ต้องทำการลดขนาดของซิมเพล็กซ์ลงด้วยการแทนที่จุดยอด $\mathbf{x}_i$ ใดๆของซิมเพล็กซ์ด้วยจุดยอดใหม่

$$\mathbf{x}_i = 0.5(\mathbf{x}_i + \mathbf{x}_l) \quad (4.19)$$

รายละเอียดของกระบวนการคำนวณของวิธีซิมเพล็กซ์ลงเขาแสดงในขั้นตอนวิธีที่ 2.6 ตัวดำเนินการหลักของวิธีหาค่าเหมาะที่สุดนี้คือการสะท้อนซึ่งถูกดำเนินการในรอบของการคำนวณ ส่วนตัวดำเนินการแบบการขยายและการหดตัวรวมไปถึงการลดขนาดของซิมเพล็กซ์จะถูกดำเนินการตามเงื่อนไขของค่าฟังก์ชันจุดประสงค์ของผลเฉลยที่ได้จากการสะท้อน เงื่อนไขในการหยุดการค้นหาคำตอบของวิธีการนี้มาหลายรูปแบบเช่น หยุดเมื่อค่าฟังก์ชันจุดประสงค์ของจุดยอดของซิมเพล็กซ์มีค่าใกล้เคียงกันมาก หรือหยุดเมื่อระยะห่างระหว่างจุดยอดของซิมเพล็กซ์มีค่าน้อยมาก กระบวนการคำนวณของวิธีซิมเพล็กซ์ลงเขา แสดงได้ดังนี้

- 1: ขั้นตอนเริ่มต้น กำหนดเซตของจุดยอดของซิมเพล็กซ์ $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n+1}\}$ และ $k = 0$
- 2: หา $\mathbf{x}_h$ และ $\mathbf{x}_l$
- 3: การสะท้อน: หา $\mathbf{x}_r$ โดยใช้สมการ (4.15)
- 4: ถ้าเงื่อนไข $f(\mathbf{x}_r) < f(\mathbf{x}_l)$ เป็นจริง ไปยังขั้นตอนที่ 8
- 5: ถ้าเงื่อนไข $f(\mathbf{x}_l) \leq f(\mathbf{x}_r) < f(\mathbf{x}_h)$ เป็นจริง แทนที่ $\mathbf{x}_h$ ด้วย $\mathbf{x}_r$ และข้ามไปยังขั้นตอนที่ 14
- 6: ถ้าเงื่อนไข $f(\mathbf{x}_r) = f(\mathbf{x}_h)$ เป็นจริง แทนที่ จุดที่ให้ค่าฟังก์ชันจุดประสงค์สูงสุดอันดับที่สองด้วย $\mathbf{x}_r$ และข้ามไปยังขั้นตอนที่ 14
- 7: ถ้าเงื่อนไข $f(\mathbf{x}_r) > f(\mathbf{x}_h)$ เป็นจริง ไปยังขั้นตอนที่ 11
- 8: การขยาย: หา $\mathbf{x}_e$ โดยใช้สมการ (4.17)
- 9: ถ้าเงื่อนไข $f(\mathbf{x}_e) < f(\mathbf{x}_r)$ เป็นจริง แทนที่ $\mathbf{x}_h$ ด้วย $\mathbf{x}_e$ และไปยังขั้นตอนที่ 14
- 10: ถ้าเงื่อนไขในขั้นตอนที่ 9 ไม่เป็นจริง ข้ามไปยังขั้นตอนที่ 14
- 11: การหดตัว: หา $\mathbf{x}_c$ โดยใช้สมการ (4.18)
- 12: ถ้าเงื่อนไข $f(\mathbf{x}_c) < f(\mathbf{x}_h)$ เป็นจริง แทนที่ $\mathbf{x}_h$ ด้วย $\mathbf{x}_c$ และไปยังขั้นตอนที่ 14
- 13: ถ้าเงื่อนไขในขั้นตอนที่ 12 ไม่เป็นจริง ลดขนาดของซิมเพล็กซ์ลงโดยใช้สมการ (4.19)
- 14: ถ้าเงื่อนไขในการหยุดค้นหาคำตอบเป็นจริง ให้หยุดการคำนวณ ถ้าไม่เป็นจริง กำหนดให้ $k = k+1$ และกลับไปยังขั้นตอนที่ 2

4.5.2 การผสมขั้นตอนวิธีวิวัฒนาการกับวิธีซิมเพล็กซ์ลงเขา

การผสมผสานขั้นตอนวิธีวิวัฒนาการกับวิธีซิมเพล็กซ์ลงเขา สามารถทำได้ในรูปแบบคล้ายกับการผสมวิธีวิวัฒนาการกับตัวดำเนินการอื่นๆ ดังที่ผ่านมา นั่นคือ กระบวนการหลักสำหรับหาผลเฉลยที่เหมาะสมที่สุด คือ ขั้นตอนวิธีวิวัฒนาการ ส่วนวิธีซิมเพล็กซ์ลงเขาจะถูกใช้เป็นตัวช่วยในการผลิตสมาชิกของประชากรลูกจำนวนหนึ่งในแต่ละวงวนของขั้นตอนวิธีวิวัฒนาการนั้นๆ ส่วนกลุ่มของผลเฉลยเริ่มต้นที่ใช้ในการสร้างซิมเพล็กซ์เริ่มต้นสามารถคัดเลือกได้จากเวกเตอร์ตัวแปรออกแบบที่เป็นสมาชิกในประชากรที่รอบของการคำนวณนั้นๆ

การออกแบบชิ้นส่วนทางกลและชิ้นส่วนยานยนต์

5.1 ความนำ

เนื้อหาที่ผ่านมาเป็นการวิจัยเกี่ยวกับการพัฒนาขั้นตอนวิธีวิวัฒนาการใหม่และหลักการเพิ่มสมรรถนะในการหาคำตอบของขั้นตอนวิธีวิวัฒนาการใหม่และวิธีวิวัฒนาการเดิมที่มีอยู่แล้ว ในบทนี้เป็นการวิจัยที่มุ่งเป้าไปที่การประยุกต์ใช้กระบวนการออกแบบที่พัฒนามาแล้วกับปัญหาการออกแบบระบบทางวิศวกรรมที่ใช้งานจริง และใช้เวลามากในการหาค่าฟังก์ชันสำหรับการออกแบบ ในที่นี้ จะนำเสนอการสาธิตการออกแบบชิ้นส่วนทางกลที่เรียกว่าแขนรับโมเมนต์บิดหรือ torque arm การออกแบบชิ้นส่วนเชื่อมต่อกระปุกเกียร์ในรถยนต์ และการออกแบบชิ้นส่วนทางกลที่มีความซับซ้อน

5.2 การออกแบบแขนรับโมเมนต์บิดด้วยการผสม SPEA2 กับแบบจำลองทดแทน

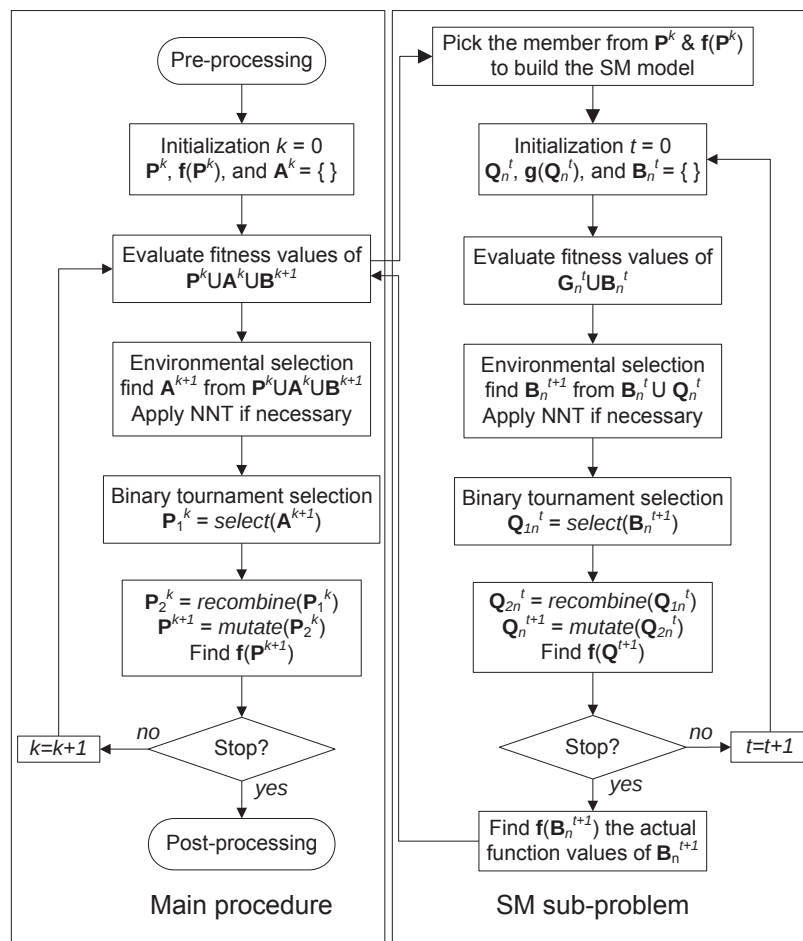
การศึกษาการออกแบบชิ้นส่วนทางกลที่เรียกว่าแขนรับโมเมนต์บิดนี้ เป็นการศึกษาการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบผสม ซึ่งในที่นี้คือการใช้ SPEA2 ผสมกับแบบจำลองทดแทน เพื่อใช้ในการออกแบบชิ้นส่วนงานจริง ชิ้นส่วนรับโมเมนต์บิดเป็นชิ้นส่วนที่ใช้ในยานยนต์เพื่อเป็นตัวเชื่อมระหว่างโครงรถกับระบบขับเคลื่อนของรถ หรืออาจนำไปใช้ในงานอื่นที่มีเงื่อนไขการใช้งานในลักษณะคล้ายกัน ในส่วนของวิธีวิวัฒนาการแบบผสม กระบวนการหาผลเฉลยที่เหมาะสมที่สุดหลักคือ SPEA2 โดยมีแบบจำลองทดแทนเป็นตัวดำเนินการย่อยเพื่อช่วยในการผลิตประชากรลูกที่มีคุณภาพสูงให้กับกระบวนการหลักของ SPEA2 ในทุกรอบของการคำนวณ

5.2.1 การผสม SPEA2 กับแบบจำลองทดแทน

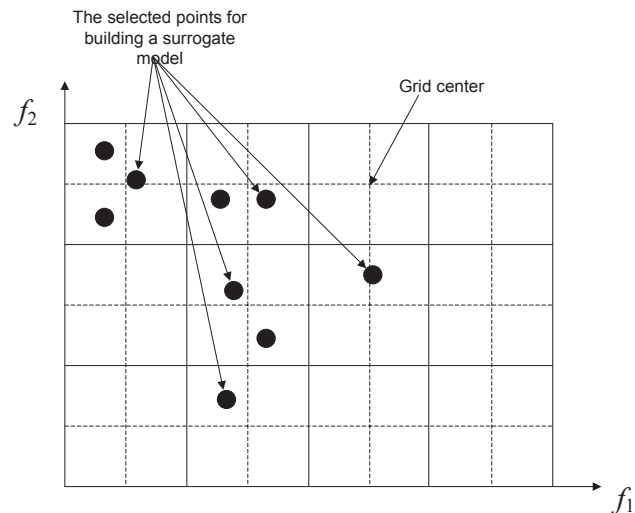
รูปที่ 5-1 แสดงการเชื่อมต่อระหว่างขั้นตอนวิธีวิวัฒนาการหลัก SPEA2 กับการหาค่าที่เหมาะสมที่สุดแบบหลายเป้าหมายโดยใช้แบบจำลองทดแทน ลักษณะของการเชื่อมต่อจะเป็นรูปแบบการเชื่อมต่อง่ายๆแสดงในหัวข้อย่อ 3.3.3 กล่าวคือ ในทุกวงวนของการค้นหาผลเฉลยที่เหมาะสมที่สุดโดย SPEA2 จะมีการผลิตประชากรลูกจากกระบวนการหาค่าที่เหมาะสมที่สุดด้วยแบบจำลองทดแทนทุกรอบของการคำนวณหรือวงวนจากรูป กระบวนการค้นหาผลเฉลยของ SPEA2 เริ่มต้นด้วยประชากรและเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอบงำตั้งต้น หลังจากนั้นทำการผลิตประชากรลูกด้วยการดำเนินการ crossover และ mutation พร้อมทั้งหาค่าฟังก์ชันจริงของผลเฉลยเหล่านั้น (ในที่นี้ SPEA2 ใช้เวกเตอร์ตัวแปรออกแบบที่เป็นจำนวนจริง) เมื่อมีกลุ่มของผลเฉลย เราสามารถสร้างแบบจำลองทดแทน และทำการประยุกต์ใช้ SPEA2 หาผลเฉลยที่เหมาะสมที่สุดแบบหลายเป้าหมายภายใต้แบบจำลองทดแทนนั้น เมื่อได้ผลเฉลยที่ไม่ถูกรอบงำจากขั้นตอนย่อยนี้ ทำการหาค่าฟังก์ชันจริงของผลเฉลยเหล่านั้น และนำมารวมกับประชากรลูกที่ผลิตโดยตัวดำเนินการของ

SPAE2 และสมาชิกในเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอบงำในวงวนก่อนหน้านี้ จากนั้นทำการหาผลเฉลยที่ไม่ถูกรอบงำของกลุ่มผลเฉลยที่รวมกันนี้ เพื่อเก็บเป็นผลเฉลยที่ไม่ถูกรอบงำสำหรับรอบของการคำนวณถัดไป

ในกรณีที่สมาชิกในเมตริกซ์เก็บผลเฉลยมีจำนวนมากเกินไป อาจส่งผลให้กระบวนการคำนวณมีความล่าช้า เราสามารถคัดผลเฉลยที่ไม่ถูกรอบงำในเมตริกซ์เก็บผลเฉลยนี้ทิ้ง ด้วยวิธี nearest neighbourhood เมื่อการหาค่าเหมาะที่สุดดำเนินการผ่านไประยะเวลาหนึ่ง จำนวนของผลเฉลยที่สามารถใช้ในการสร้างแบบจำลองทดแทนจะมีจำนวนมากขึ้น ส่งผลให้คอมพิวเตอร์ต้องใช้หน่วยความจำสูงขึ้น เพื่อแก้ปัญหานี้ เราประยุกต์ใช้วิธี adaptive grid เพื่อคัดสมาชิกบางส่วนทิ้ง การคัดสรรในรูปแบบนี้จะเป็นการคัดสรรที่โดเมนของฟังก์ชันเป้าหมาย ดังแสดงในรูปที่ 5-2



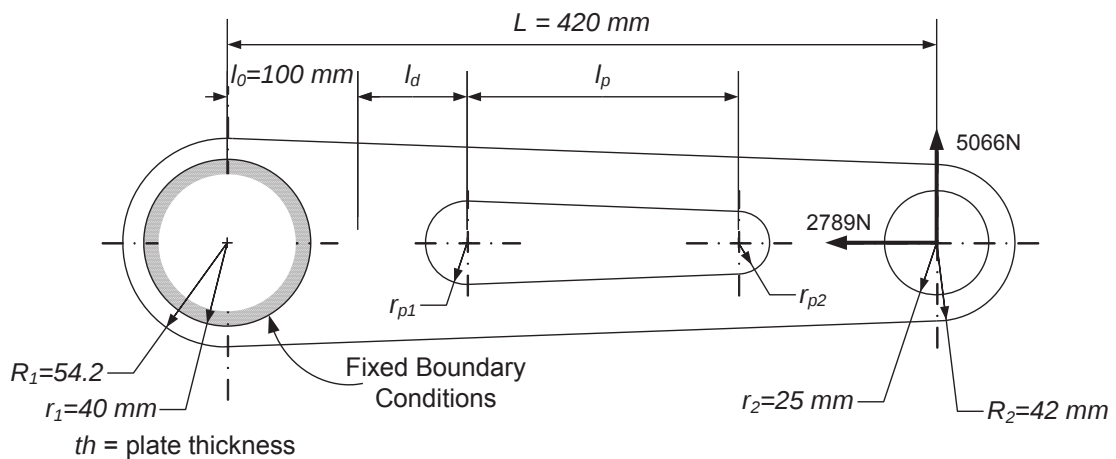
รูปที่ 5-1 flowchart ของการผสม SPEA2 กับแบบจำลองทดแทน



รูปที่ 5-2 วิธี adaptive grid สำหรับคัดสรรผลเฉลยสำหรับการสร้างแบบจำลองทดแทน

5.2.2 ปัญหาการออกแบบ

โครงสร้างของแขนรับโมเมนต์บิดแสดงในรูปที่ 5-3 เป็นโครงสร้างที่ทำจากโลหะแผ่นเจาะรูกลมหัวท้าย รูด้านซ้ายมีเงื่อนไขขอบเขตสำหรับการวิเคราะห์ทางไฟไนต์เอลิเมนต์ ส่วนรูด้านขวารับภาระในแนวนอนและแนวตั้งดังรูป จากเงื่อนไขการใช้งาน โครงสร้างมีโอกาสเกิดการเสียหายจากความเค้น การโก่งตัว และในกรณีของภาระแบบพลวัต ค่าความถี่ธรรมชาติของโครงสร้างหรืออีกนัยหนึ่งคือค่าความแข็งดิ่งเชิงพลวัต (dynamic stiffness) ควรจะมีค่าสูงพอที่จะรับแรงพลวัตที่มีความถี่สูงๆ และไม่เกิดการกำทอน (resonance) พารามิเตอร์ที่กำหนดให้เป็นค่าคงที่มีดังนี้ $l_d = 30$ mm, $l_p = 190$ mm, $r_{p1} = 26$ mm, และ $r_{p2} = 27$ mm ส่วนตัวแปรออกแบบและขอบเขตของตัวแปรแสดงในตารางที่ 5-1



รูปที่ 5-3 แขนรับโมเมนต์บิด

ปัญหาการออกแบบในการศึกษานี้มี 6 ปัญหาซึ่งเขียนแทนด้วย F1, F2, F3, F4, F5, และ F6 ดังนี้

Objective function: $\min f_1(\mathbf{x})$ & $\max f_2(\mathbf{x})$

F1: $f_1(\mathbf{x}) = w$ & $f_2(\mathbf{x}) = \omega$

F2: $f_1(\mathbf{x}) = w$ & $f_2(\mathbf{x}) = \lambda$

$$F3: f_1(\mathbf{x}) = w \text{ \& } f_2(\mathbf{x}) = \lambda + \omega/\omega_0$$

Subject to

$$\sigma_{von} - \sigma_y/S_F \leq 0$$

$$1 - \lambda \leq 0$$

Objective function: $\min f_1(\mathbf{x}) \text{ \& } \max f_2(\mathbf{x})$

$$F4: f_1(\mathbf{x}) = w \text{ \& } f_2(\mathbf{x}) = \omega$$

$$F5: f_1(\mathbf{x}) = w \text{ \& } f_2(\mathbf{x}) = \lambda$$

$$F6: f_1(\mathbf{x}) = w \text{ \& } f_2(\mathbf{x}) = \lambda + \omega/\omega_0$$

Subject to

$$\sigma_{von} - \sigma_y/S_F \leq 0$$

$$1 - \lambda \leq 0$$

$$\omega - \omega_0 \leq 0$$

เมื่อ w = น้ำหนักโครงสร้าง, ω = ความถี่ธรรมชาติของโครงสร้างโหมดที่ 1, λ = ค่าปัจจัยการโก่งตัว (อัตราส่วนระหว่างแรงวิกฤตกับแรงกระทำจริง), σ_{von} = ความเค้น von Mises สูงสุดที่เกิดกับโครงสร้าง, และ $\omega_0 = 7.2688$ rad/s ค่าโมดูลัสของวัสดุที่ใช้ทำชิ้นส่วนนี้คือ $E = 60 \times 10^6$ N/cm², ค่าอัตราส่วนพัวของ $\nu = 0.3$, ค่าความเค้นครากคือ $\sigma_y = 80000$ N/cm², ความหนาแน่นของวัสดุ $\rho = 0.00718$ kg/cm³, และค่าความปลอดภัย $S_F = 1.5$.

ตารางที่ 5-1 ตัวแปรออกแบบและขอบเขตบน-ล่าง

Design variables	Bounds (mm)
$x_1 = l_d$	$0 \leq l_d \leq 60$
$x_2 = l_p$	$100 \leq l_p \leq 200$
$x_3 = r_{p1}$	$15 \leq r_{p1} \leq 35$
$x_4 = r_{p2}$	$15 \leq r_{p2} \leq 30$
$x_5 = th$	$3 \leq th \leq 4.5$

ปัญหาการออกแบบ F1, F2, F3 มีความคล้ายคลึงกับปัญหา F4, F5, F6 ตามลำดับยกเว้นในปัญหาการออกแบบสามอันหลังมีการเพิ่มเงื่อนไขบังคับกับความถี่ธรรมชาติเข้าไป เนื่องจากการศึกษาในหัวข้อนี้เป็นการศึกษาสมรรถนะของ SPEA2 ที่ใช้แบบจำลองทดแทนรูปแบบต่างๆกัน ดังนั้นขั้นตอนวิธีวิวัฒนาการแบบผสมที่ใช้ในการหาเซตของผลเฉลยที่เหมาะสมที่สุดของพารามิเตอร์ มีดังต่อไปนี้

- SPEA-O หมายถึง SPEA2 ที่ไม่ใช้แบบจำลองทดแทน
- SPEA-Q หมายถึง SPEA2 ที่ใช้ฟังก์ชันกำลังสองเป็นแบบจำลองทดแทน
- SPEA-R หมายถึง SPEA2 ที่ใช้ฟังก์ชันฐานหลักแนวรัศมีเป็นแบบจำลองทดแทน
- SPEA-N หมายถึง SPEA2 ที่ใช้โครงข่ายประสาทเทียมเป็นแบบจำลองทดแทน
- SPEA-K หมายถึง SPEA2 ที่ใช้แบบจำลอง Kriging เป็นแบบจำลองทดแทน

ขั้นตอนวิธีวิวัฒนาการแต่ละวิธีถูกใช้หาผลเฉลยของปัญหาการออกแบบใดๆ 10 ครั้ง เพื่อสามารถวัดสมรรถนะในการหาผลเฉลยของวิธีหาค่าที่เหมาะสมที่สุดเหล่านั้นได้ ขนาดของประชากรในแต่ละวงวนคือ 20

และจำนวนวงวนสูงสุดที่ใช้คือ 30 ในกรณีของ SPEA-O กรณีของขั้นตอนวิธีแบบผสม ใช้ขนาดประชากรเท่ากันคือ 20 และในแต่ละรอบของการคำนวณ SPEA2 ที่ใช้แบบจำลองทดแทนผลิตจำนวนประชากรลูกเท่ากับ 20 นั้นหมายความว่า ถ้าต้องการใช้จำนวนของการวิเคราะห์ทางไฟไนท์เอลิเมนต์ของโครงสร้างเท่ากับกรณีของ SPEA2 ที่ไม่ใช้แบบจำลองทดแทน จำนวนวงวนสูงสุดต้องมีค่าเป็น 15 รอบ อัตราการ crossover และ mutation มีค่าเป็น 1.0 และ 0.1 ตามลำดับ ขนาดสูงสุดของเมตริกซ์เก็บผลเฉลยคือ 100

5.2.3 ผลการออกแบบและการเปรียบเทียบสมรรถนะ

จากการรันวิธีวิวัฒนาการเพื่อหาผลเฉลยของปัญหาออกแบบทั้งหมด วิธีละ 10 ครั้งในแต่ละปัญหา ทำให้เราได้แนวของพารेटโต 10 แนวสำหรับวิธีวิวัฒนาการใดๆ จากนั้นทำการคำนวณค่าไฮเปอร์วอลุ่ม (hypervolume, HV) ของแนวพารेटโตทั้ง 10 ค่า ผลการเปรียบเทียบสมรรถนะของวิธีวิวัฒนาการต่างๆ ในการค้นหาผลเฉลยพารेटโตของปัญหาการออกแบบทั้ง 6 แสดงในตารางที่ 5-2 ค่าต่างๆในตารางคือค่าเฉลี่ยของปริมาตรของแนวพารेटโตที่ถูกปรับค่า โดยค่าเฉลี่ยสูงสุดถูกปรับเป็น 1 และค่าเฉลี่ยต่ำสุดถูกปรับเป็น 0 เพื่อความง่ายในการเปรียบเทียบ โดยหลักการเปรียบเทียบปริมาตรของแนวพารेटโต วิธีที่ให้ค่าปริมาตรสูงสุดคือวิธีที่มีสมรรถนะสูงสุด

จากตาราง วิธีที่ดีที่สุดสำหรับปัญหา F1 คือ SPEA-R และวิธีที่ดีอันดับสองคือ SPEA-Q ขั้นตอนวิธี SPEA-Q ให้ผลดีที่สุดในการกรณีของ F2 ในขณะที่วิธีที่ดีอันดับรองลงมาคือ SPEA-N ส่วนวิธีวิวัฒนาการที่ดีที่สุดสำหรับ F3, F4, F5, และ F6 คือ SPEA-K, SPEA-R, SPEA-K, และ SPEA-K ตามลำดับ ส่วนวิธีที่รองลงมาสำหรับปัญหาทั้ง 4 คือ SPEA-Q ขั้นตอนวิธี SPEA-O เป็นวิธีที่มีสมรรถนะต่ำที่สุดสำหรับปัญหาการออกแบบ F2, F5, และ F6 ส่วนปัญหาการออกแบบที่เหลือวิธีที่ประสิทธิภาพต่ำสุดคือ SPEA-N โดยภาพรวม วิธีที่ดีที่สุดคือ SPEA-Q รองลงมาคือ SPEA-K ดังผลการเปรียบเทียบในตาราง

ตารางที่ 5-2 ผลการเปรียบเทียบสมรรถนะของขั้นตอนวิธีวิวัฒนาการด้วยไฮเปอร์วอลุ่ม

	SPEA-O	SPEA-Q	SPEA-R	SPEA-N	SPEA-K
F1	0.429	0.704	1.000	0.000	0.523
F2	0.000	1.000	0.052	0.917	0.692
F3	0.577	0.995	0.745	0.000	1.000
F4	0.457	0.916	1.000	0.000	0.497
F5	0.000	0.976	0.640	0.161	1.000
F6	0.000	0.673	0.243	0.007	1.000
Total	1.463	5.264	3.679	1.084	4.713

การเปรียบเทียบเชิงสถิติในรูปแบบเดียวกับตารางที่ 4-3 และ 4-4 แสดงในตารางที่ 5-3 และ 5-4 ตามลำดับ การทดสอบนี้ใช้วิธี t-test ที่ระดับความเชื่อมั่น 95% ผลของคะแนนแสดงในตารางที่ 5-4 ในที่นี้วิธีที่ดีที่สุดจะมีคะแนนน้อยสุด จากตารางพบว่าวิธีที่มีสมรรถนะสูงสุดคือ SPEA-Q รองลงมาคือ SPEA-K ซึ่งไม่มีความแตกต่างกันมากนัก ส่วนวิธีที่ประสิทธิภาพต่ำสุดคือ SPEA-N ซึ่งประยุกต์ใช้โครงข่ายประสาทเทียม

ตารางที่ 5-3 การเปรียบเทียบวิธีต่างๆเชิงสถิติของปัญหาการออกแบบ F1

	SPEA-O	SPEA-Q	SPEA-R	SPEA-N	SPEA-K
SPEA-O	0	0	1	0	0
SPEA-Q	0	0	1	0	0
SPEA-R	0	0	0	0	0
SPEA-N	1	1	1	0	1
SPEA-K	0	0	1	0	0
Total	1	1	4	0	1
Ranking	2	2	1	5	2

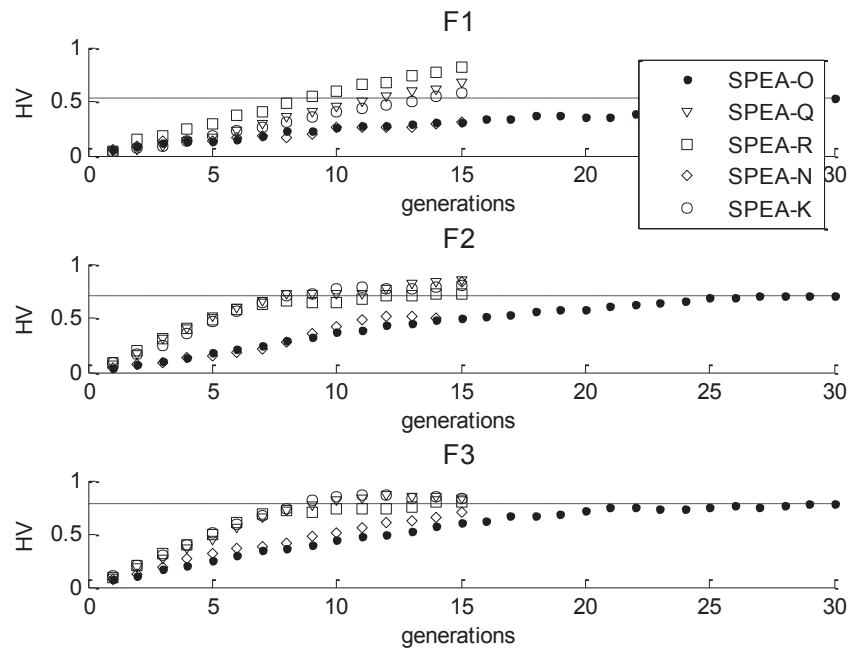
ตารางที่ 5-4 การเปรียบเทียบวิธีต่างๆเชิงสถิติของปัญหาการออกแบบทั้ง 6

	SPEA-O	SPEA-Q	SPEA-R	SPEA-N	SPEA-K
F1	2	2	1	5	2
F2	4	1	4	1	1
F3	1	1	1	5	1
F4	3	1	1	5	3
F5	4	1	3	4	1
F6	3	2	3	3	1
Total	17	8	13	23	9

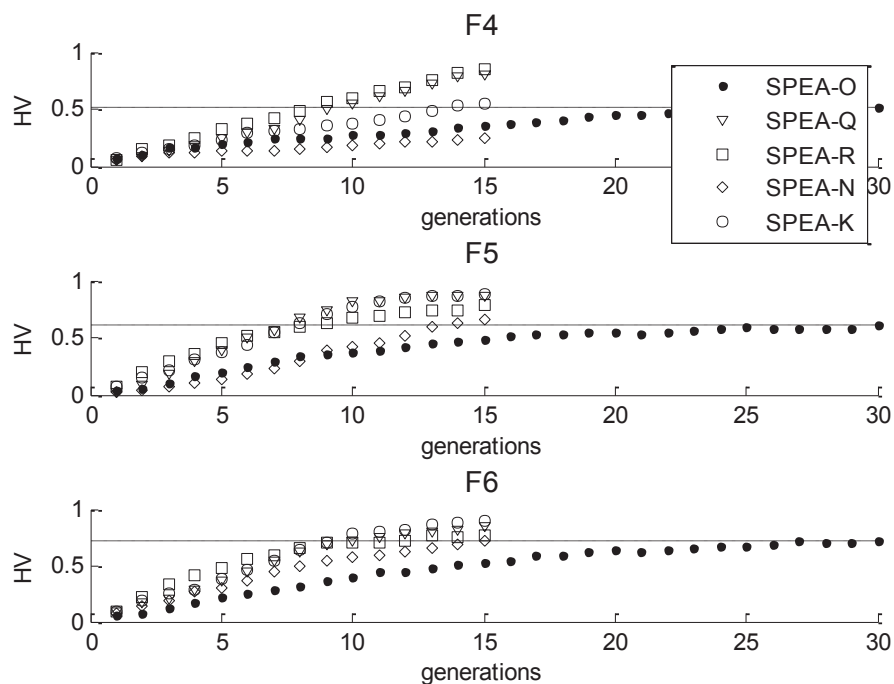
รูปที่ 5-4 แสดงประวัติการค้นหาค่าเฉลยของปัญหา F1, F2 และ F3 ด้วยขั้นตอนวิธีวิวัฒนาการแบบต่างๆ ส่วนประวัติการค้นหาคำตอบของปัญหาการออกแบบ F4, F5 และ F6 แสดงในรูปที่ 5-5 จากรูปทั้งสองพบว่า SPEA-Q และ SPEA-K จะใช้จำนวนครั้งในการวิเคราะห์ทางไฟไนต์เอลิเมนต์โดยประมาณเพียงครึ่งหนึ่งของจำนวนการวิเคราะห์ทางไฟไนต์เอลิเมนต์โดยวิธีวิวัฒนาการ SPEA-O ซึ่งเป็นวิธีที่ไม่ใช้แบบจำลองทดแทน ส่วนวิธี SPEA-R มีสมรรถนะสูงในกรณีของปัญหาการออกแบบ F1, F4 และ F5 อย่างไรก็ตามในกรณีของ SPEA-N ผลการหาคำตอบที่เหมาะสมที่สุดไม่มีความแตกต่างจากการประยุกต์ใช้ SPEA2 โดยไม่มีแบบจำลองทดแทน เราอาจสรุปได้ว่า สำหรับปัญหาการออกแบบโครงสร้างนี้โดยรวมแบบจำลองทดแทน Kriging และฟังก์ชันกำลังสอง เป็นแบบจำลองที่มีประสิทธิภาพและประสิทธิภาพสูงสุด ส่วนแบบจำลองฟังก์ชันฐานหลักแนวรัศมี มีประโยชน์ในบางกรณีของปัญหาการออกแบบ ส่วนแบบจำลองที่ใช้โครงข่ายประสาทเทียมมีประสิทธิภาพต่ำที่สุด

เพื่อหาข้อสรุปเพิ่มเติม ได้ทำการวิเคราะห์ความแม่นยำของการประมาณค่าฟังก์ชันในการออกแบบนี้ด้วยแบบจำลองทดแทนต่างๆ โดยมีชุดการทดลองสองชุดดังนี้ ชุดที่หนึ่ง ทำการสุ่มเวกเตอร์ตัวแปรออกแบบขึ้นมาจำนวนหนึ่งด้วย LHS และสร้างเซตของผลเฉลยกลุ่มที่สองด้วยวิธีเดียวกัน จากนั้นหาค่าฟังก์ชันจริงของเวกเตอร์ผลเฉลยทั้งสองกลุ่ม เวกเตอร์ผลเฉลยกลุ่มแรก ใช้ในการสร้างแบบจำลองทดแทนต่างๆ ส่วนเวกเตอร์ผลเฉลยกลุ่มที่สองใช้ในการทำนายความคลาดเคลื่อนของแบบจำลองทดแทนเทียบกับค่าจริง ตัวบ่งชี้ความแม่นยำของแบบจำลองทดแทนคือค่า root mean square error (RMSE) ผลการเปรียบเทียบแสดงในตารางที่ 5-5 จากตารางพบว่า แบบจำลองทดแทนที่มีความแม่นยำที่สุดโดยภาพรวมคือฟังก์ชันกำลังสอง รองลงมาคือแบบจำลอง Kriging ซึ่งให้ผลสอดคล้องกับผลการเปรียบเทียบสมรรถนะของขั้นตอนวิธีวิวัฒนาการก่อนหน้านี้ ตารางที่ 5-6 เป็นการศึกษากรณีคล้ายกับการเปรียบเทียบในตารางที่ 5-5 ข้อแตกต่าง

คือ ตารางที่ 5-6 เลือกผลเฉลยสองกลุ่มที่แตกต่างกันมาจาก กลุ่มเวกเตอร์ตัวแปรออกแบบที่อยู่ใกล้แนวพาเรโต กลุ่มแรก ใช้สร้างแบบจำลองทดแทนและกลุ่มที่สองใช้ในการทำนายความคลาดเคลื่อนของแบบจำลองนั้น ผลจากตาราง 5-6 ให้ข้อสรุปเช่นเดียวกับผลในตาราง 5-5 ซึ่งสามารถสรุปต่อไปอีกว่า แบบจำลองทดแทนที่มีความแม่นยำสูงจะส่งผลให้ขั้นตอนวิธีวิวัฒนาการผสมที่ใช้แบบจำลองทดแทนนั้นมีสมรรถนะในการหาผลเฉลยสูงตามไปด้วย



รูปที่ 5-4 ประวัติการค้นหาคำตอบของวิธีต่างๆ สำหรับ F1, F2, และ F3



รูปที่ 5-5 ประวัติการค้นหาคำตอบของวิธีต่างๆ สำหรับ F4, F5, และ F6

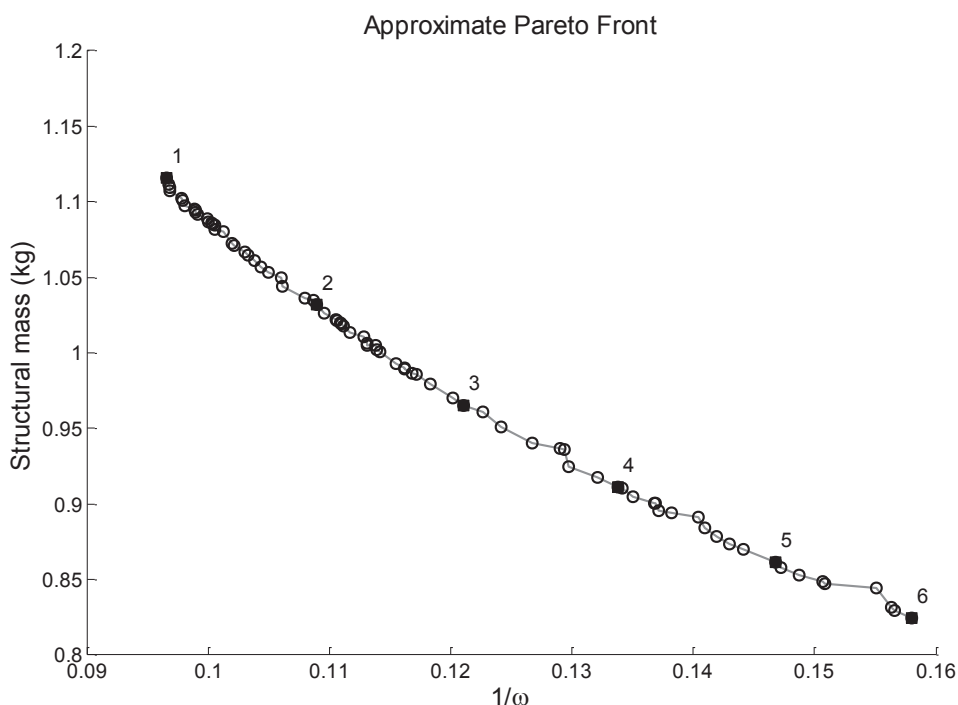
ตาราง 5-5 RMSE ของแบบจำลองทดแทน 1

Surrogate models	W	σ_{von}	λ	ω
Quadratic	0.234	8557.326	0.129	0.114
Radial-basis function	0.835	7905.801	0.365	0.534
Neural network	1.726	36085.47	0.843	1.110
Kriging	0.121	9600.842	0.202	0.198

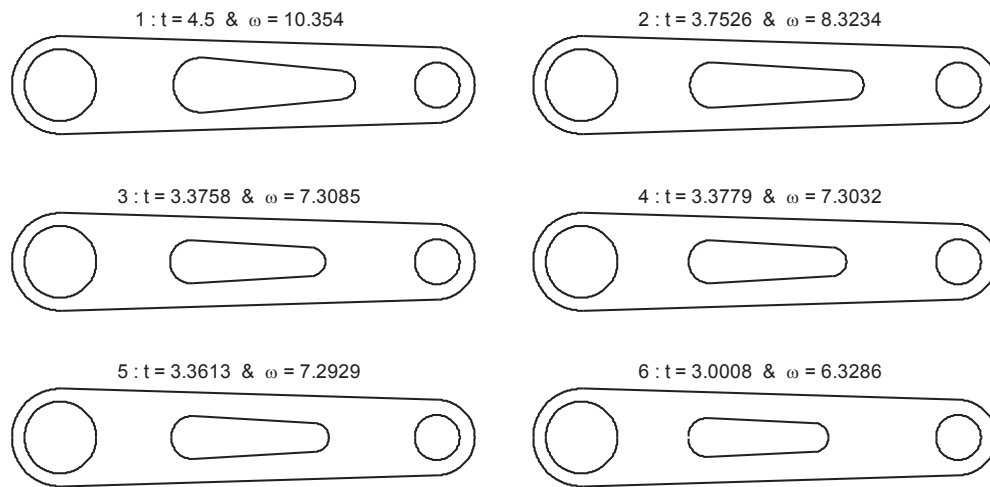
ตาราง 5-6 RMSE ของแบบจำลองทดแทน 2

Surrogate models	w	σ_{von}	λ	ω
Quadratic	0.002	170.826	0.006	0.006
Radial-basis function	0.127	685.544	0.043	0.090
Neural network	0.650	470.008	0.304	0.775
Kriging	0.027	318.878	0.011	0.017

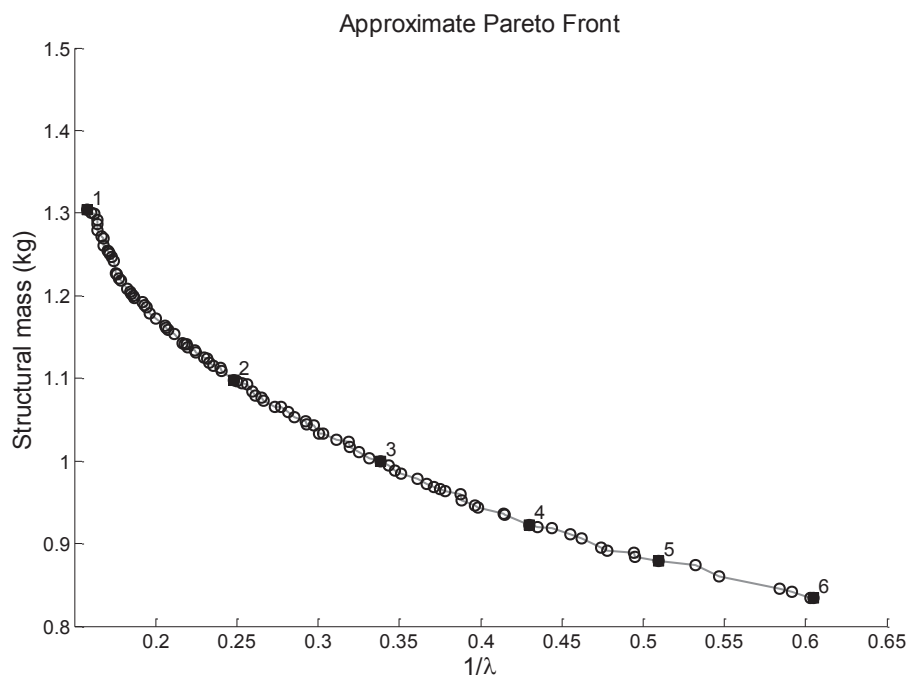
แนวของพารेटอที่ดีที่สุดที่ได้วิธีการต่างๆของปัญหาการออกแบบ F1, F2, F3, F4, F5 และ F6 แสดงในรูปที่ 5-6 5-8 5-10 5-12 5-14 และ 5-16 ตามลำดับ ส่วนรูปทรงและขนาดความหนาของชิ้นส่วนแขนรับโมเมนต์บิด ที่ได้จากแนวพารेटอเหล่านั้น แสดงในรูปที่ 5-7 5-9 5-11 5-13 5-15 และ 5-17 มีข้อสรุปเพิ่มเติมอีกอย่างหนึ่งสำหรับการศึกษาครั้งนี้คือ ในการรันวิธีวิวัฒนาการแบบหลายเป้าหมายหนึ่งครั้งทำให้ได้ผลลัพธ์ที่เหมาะสมที่สุดหลากหลายรูปแบบสำหรับเลือกใช้งาน ซึ่งเป็นประโยชน์โดยตรงของการหาค่าที่เหมาะสมที่สุดแบบหลายเป้าหมาย



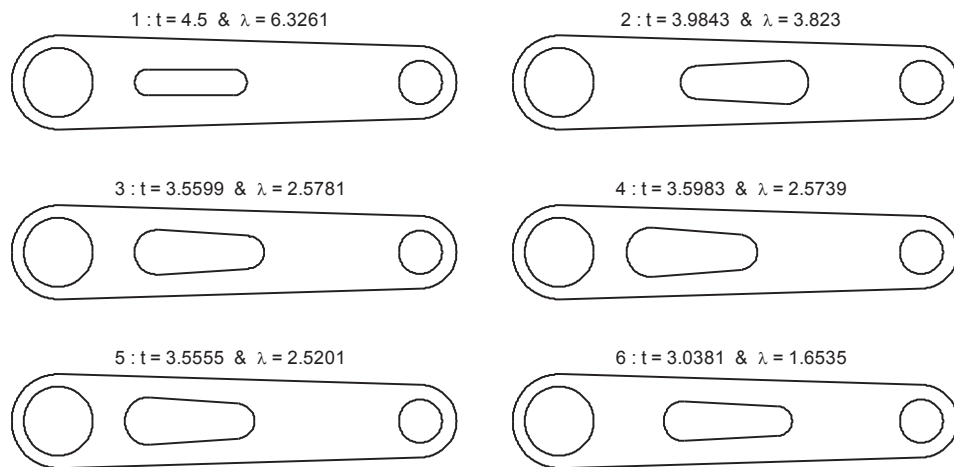
รูปที่ 5-6 แนวพารेटอของ F1



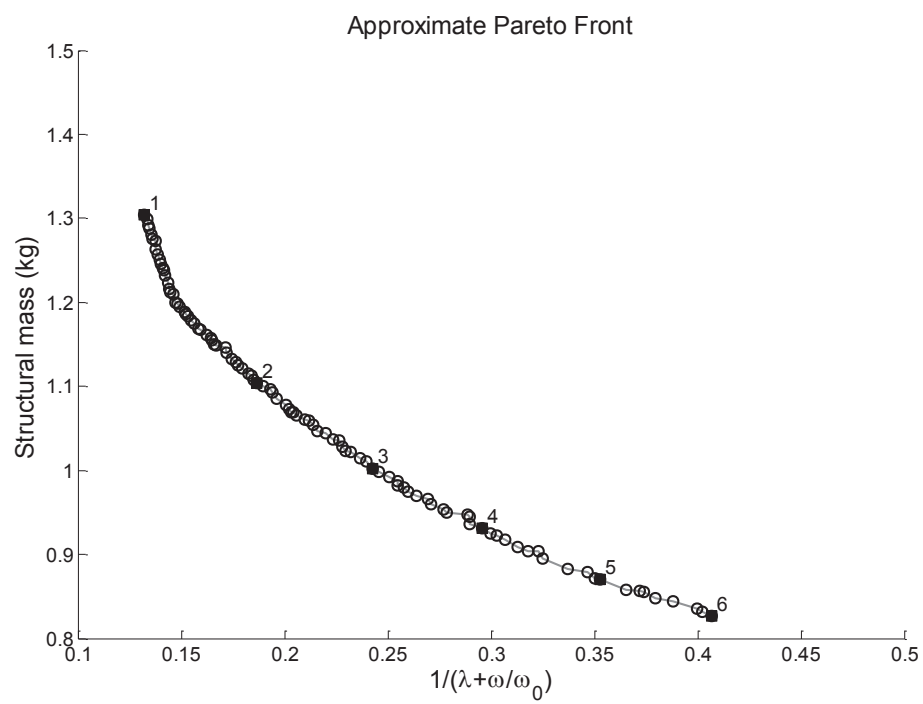
รูปที่ 5-7 ผลเฉลยจากแนวพาเรโตของ F1



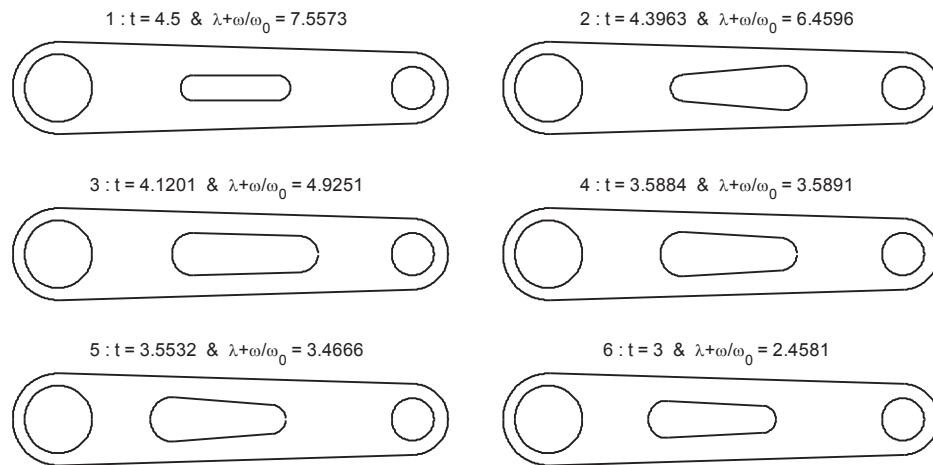
รูปที่ 5-8 แนวพาเรโตของ F2



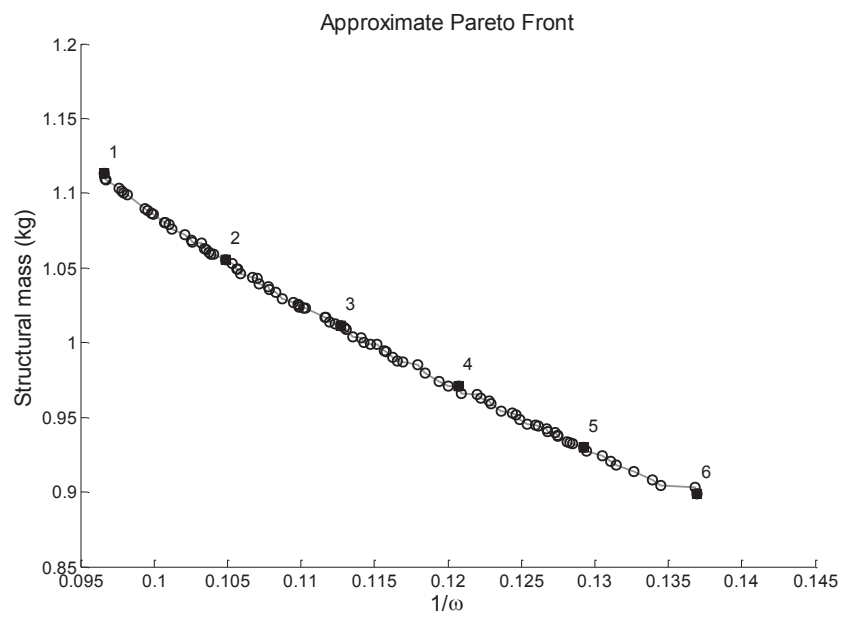
รูปที่ 5-9 ผลเฉลยจากแนวพาเรโตของ F2



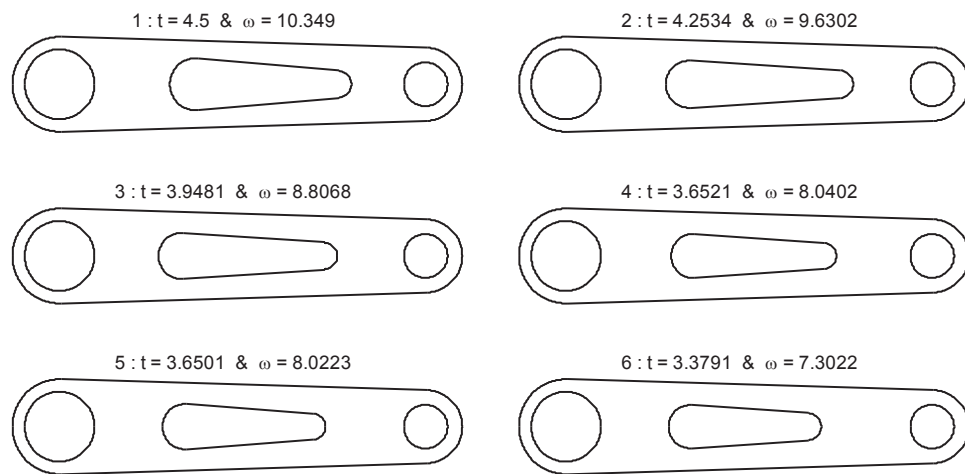
รูปที่ 5-10 แนวพาเรโตของ F3



รูปที่ 5-11 ผลเฉลยจากแนวพารโตะของ F3

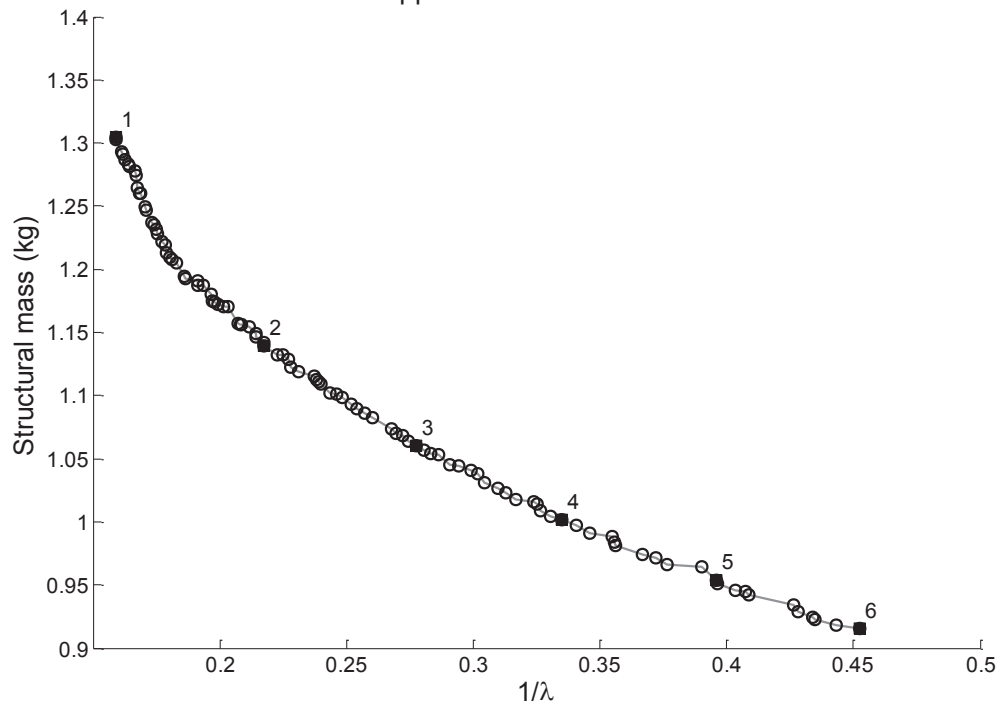


รูปที่ 5-12 แนวพารโตะของ F4

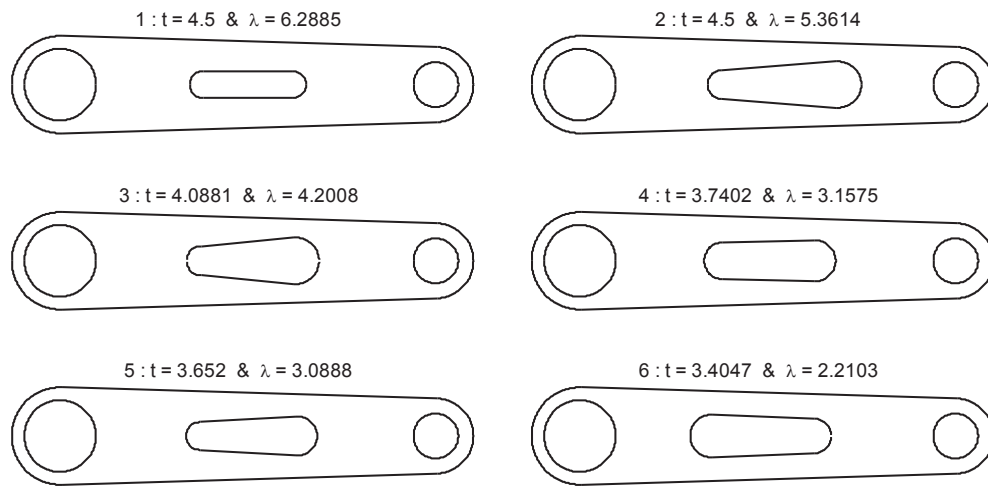


รูปที่ 5-13 ผลเฉลยจากแนวพาเรโตของ F4

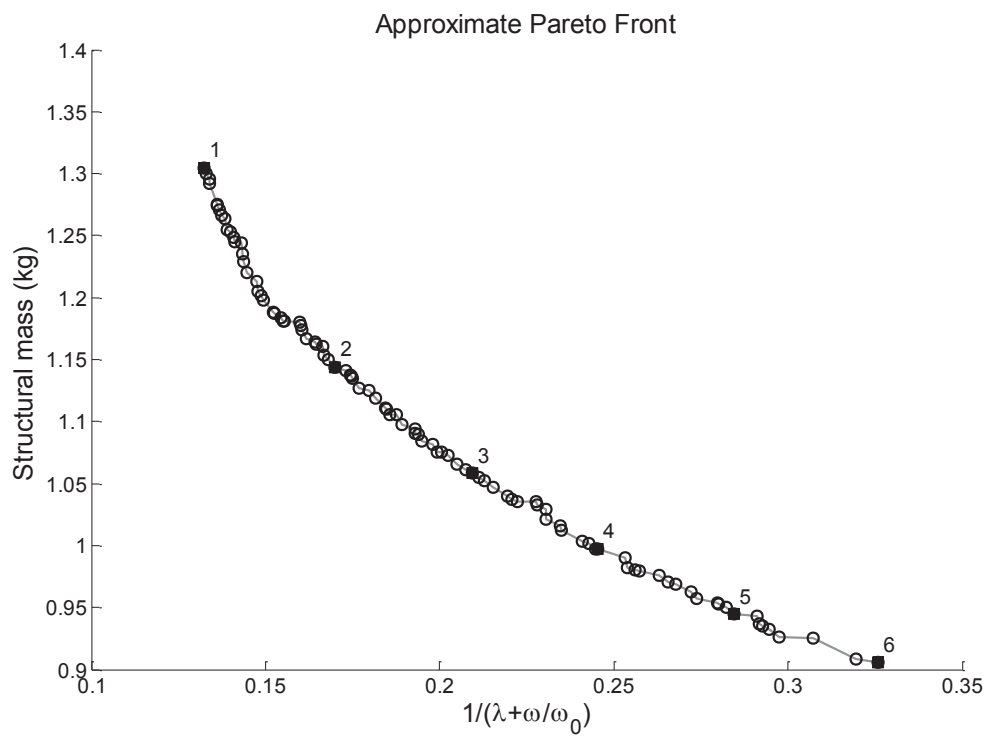
Approximate Pareto Front



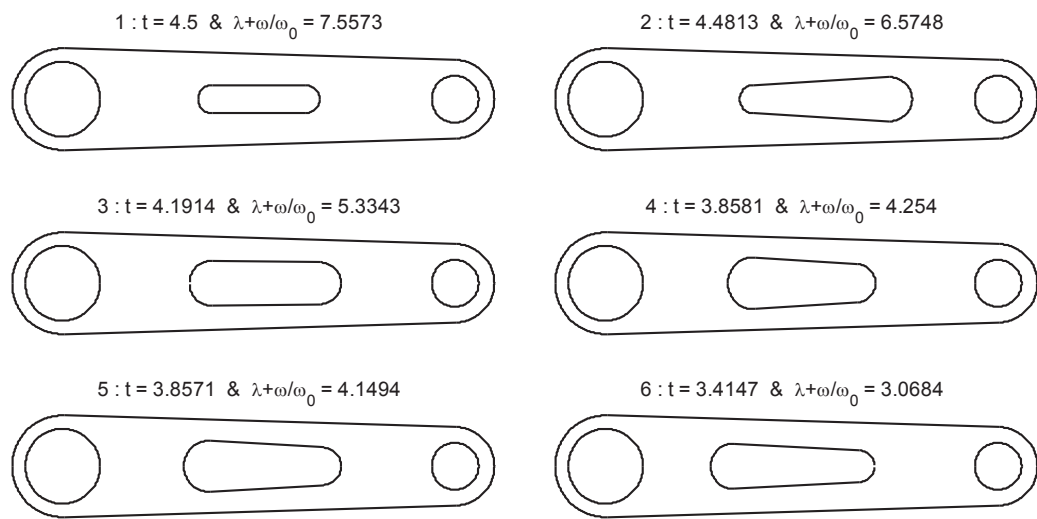
รูปที่ 5-14 แนวพาเรโตของ F5



รูปที่ 5-15 ผลเฉลยจากแนวพารโศของ F5



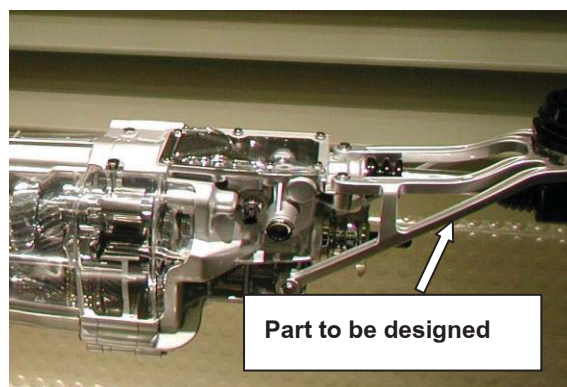
รูปที่ 5-16 แนวพารโศของ F6



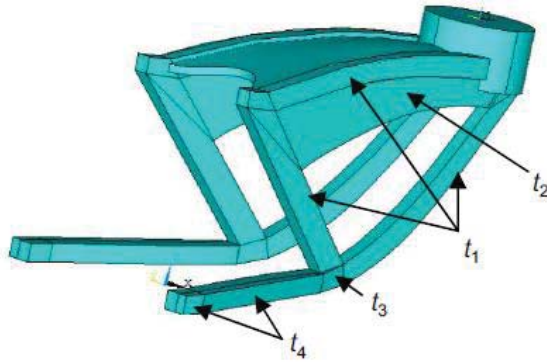
รูปที่ 5-17 ผลเฉลยจากแนวพาเรโตของ F7

5.3 การออกแบบชิ้นส่วนรองรับกระปุกเกียร์

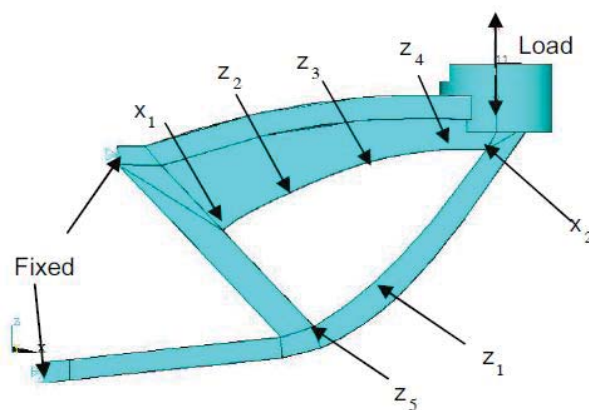
หัวข้อย่อยนี้สาธิตการออกแบบชิ้นส่วนยานยนต์ด้วยขั้นตอนวิธีวิวัฒนาการแบบผสม รูปที่ 5-18 แสดงตำแหน่งที่ติดตั้งชิ้นส่วนดังกล่าวซึ่งเป็นการเชื่อมต่อระหว่างเครื่องยนต์และโครงของรถยนต์ที่บริเวณกระปุกเกียร์ เมื่อเครื่องยนต์ทำงาน ชิ้นส่วนจะรับแรงพลวัต ดังนั้นจึงมีโอกาที่จะเกิดการเสียหายเนื่องจากการล้า ความเค้น การโก่งตัว หรือเกิดจากความไม่เสถียรเชิงพลวัต ดังนั้น ในการออกแบบชิ้นส่วนยานยนต์นี้ จำเป็นต้องคำนึงถึงปัจจัยสำคัญที่จะก่อให้เกิดความเสียหายดังที่กล่าวมาข้างต้น เนื่องจากได้มีงานวิจัยก่อนหน้านี้ได้ดำเนินการออกแบบชิ้นงานนี้ขึ้นมาใหม่และมีสมรรถนะทางกลสูงกว่าชิ้นงานเดิม (ดูรายละเอียดใน [10]) ดังนั้น ในการสาธิตการออกแบบนี้จะเลือกใช้ชิ้นงานใหม่ซึ่งแสดงในรูปที่ 5-19 และ 5-20 ตัวแปรออกแบบสำหรับกรณีนี้มีทั้งตัวแปรกำหนดขนาดและกำหนดรูปทรง (sizing and shape design variables) ของชิ้นงานดังแสดงในรูปที่ 5-19 และ 5-20



รูปที่ 5-18 ชิ้นงานที่ออกแบบ



รูปที่ 5-19 ตัวแปรกำหนดขนาด



รูปที่ 5-20 ตัวแปรกำหนดรูปทรง

สำหรับปัญหาการออกแบบ เนื่องจากชิ้นงานรับแรงพลวัตที่เกิดจากการสั่นสะเทือนของเครื่องยนต์ และแรงแบบสุ่มที่เกิดจากการที่รถเคลื่อนที่ไปตามถนน ดังนั้น ฟังก์ชันเป้าหมายที่ใช้คือการหาค่าความเค้นดึงเชิงพลวัตของชิ้นส่วนสูงที่สุด และในขณะเดียวกัน เพื่อลดราคาของชิ้นงาน จะทำการหาค่าน้ำหนักต่ำที่สุดของชิ้นงานนี้ โดยที่เงื่อนไขบังคับของปัญหาการออกแบบนี้คือ ความเค้น และการกระจัดหรือระยะยุบตัวของชิ้นงาน ปัญหาการหาค่าเหมาะที่สุดแบบหลายเป้าหมายสำหรับชิ้นส่วนนี้ สามารถเขียนได้ดังนี้

$$\text{Min: } \mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x})] \quad (5.1)$$

Subject to

$$\sigma_{\max} \leq \sigma_{\text{allowable}}$$

$$u_{\max} \leq 0.005$$

$$0.0015 \leq t_1 \leq 0.0115$$

$$0.0015 \leq t_2 \leq 0.0115$$

$$0.0015 \leq t_3 \leq 0.01$$

$$0.0025 \leq t_4 \leq 0.015$$

$$-0.003 \leq z_1 \leq 0.01$$

$$-0.0025 \leq z_2 \leq 0.0028$$

$$-0.01 \leq z_3 \leq 0.005$$

$$-0.01 \leq z_4 \leq 0.005$$

$$0 \leq z_5 \leq 0.03$$

$$-0.01 \leq x_1 \leq 0.01$$

$$-0.005 \leq x_2 \leq 0.01$$

เมื่อ $\mathbf{x}$ คือเวกเตอร์ตัวแปรออกแบบที่มีเอลิเมนต์ตามที่แสดงในรูปที่ 5-19 และ 5-20 f_1 คือมวลของชิ้นงาน f_2 คือค่าความแข็งแรงดึงของโครงสร้าง รายละเอียดของเวกเตอร์ออกแบบและฟังก์ชันเป้าหมายมีดังนี้

$$\mathbf{x} = \{t_1, t_2, t_3, t_4, z_1, z_2, z_3, z_4, z_5, x_1, x_2\}^T$$

$$f_1 = mass \quad (5.2)$$

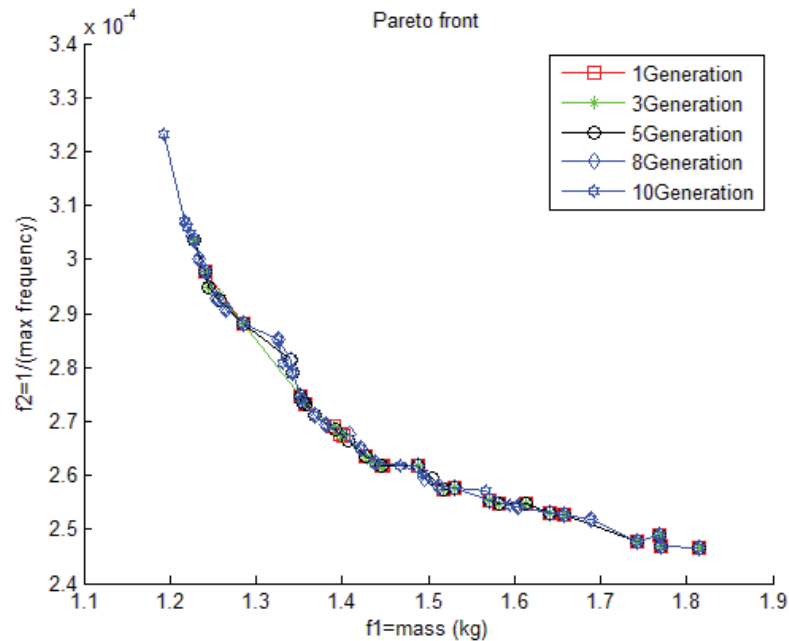
$$f_2 = \frac{1}{\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5} \quad (5.3)$$

เมื่อ σ_{max} = ค่าความเค้นแบบ von Misses สูงสุดบนชิ้นงาน $\sigma_{allowable}$ = ค่าความเค้นที่ยอมรับได้ t_i = ความหนาของชิ้นงาน z_i = จุดสำคัญในแนวแกน z สำหรับชิ้นรูปชิ้นส่วน x_i = จุดสำคัญในแนวแกน x สำหรับชิ้นรูปชิ้นส่วน ω_i = ค่าความถี่ธรรมชาติโหมดที่ i -th ของโครงสร้าง

ในแบบจำลองทางไฟไนท์เอลิเมนต์ โครงสร้างรับภาระสามรูปแบบกล่าวคือ การดัดในแนวดิ่ง การดัดในแนวสายด้านข้างและการบิด โดยภาระทั้งสามกรณีกระทำที่ปลายด้านขวาของโครงสร้าง ส่วนปลายด้านซ้ายของโครงสร้างซึ่งมี 4 จุดเชื่อมกับเครื่องยึด จะทำการประยุต์เงื่อนไขขอบเขตแบบยึดกับพื้นเพื่อความสะดวกในการวิเคราะห์ ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายที่ใช้หาผลเฉลยพาเรโตของปัญหาการออกแบบนี้คือวิธี PBIL แบบหลายเป้าหมายที่ใช้ตัวแปรเลขฐานสองในการดำเนินการหาคำตอบ ในที่นี้ จะเขียนแทนด้วย MOPBIL ในกรณีของการผสมแบบจำลองทดแทนเพื่อใช้ร่วมกับ MOPBIL นั้น เลือกใช้แบบจำลอง Kriging และรูปแบบการผสมผสานเป็นไปตามรูปแบบในหัวข้อย่อย 3.3.2 ในการสาธิตการออกแบบนี้ ประยุต์ใช้ LHS สร้างจุดของผลเฉลย 100 จุดสำหรับสร้างแบบจำลองทดแทน หลังจากนั้นใช้ MOPBIL หาผลเฉลยภายใต้การหาค่าฟังก์ชันด้วยแบบจำลองทดแทน เมื่อได้ผลเฉลยที่ไม่ถูกรอรับจากเฟสนี้แล้ว ดำเนินการค้นหาผลเฉลยที่เหมาะสมที่สุดด้วย MOPBIL แบบปกติ โดยใช้จำนวนประชากรเท่ากับ 30 และจำนวนรอบของการคำนวณคือ 10 แผนการหาผลเฉลยด้วยวิธีวิวัฒนาการแบบผสมนี้เรียกว่า MOPBIL-SM

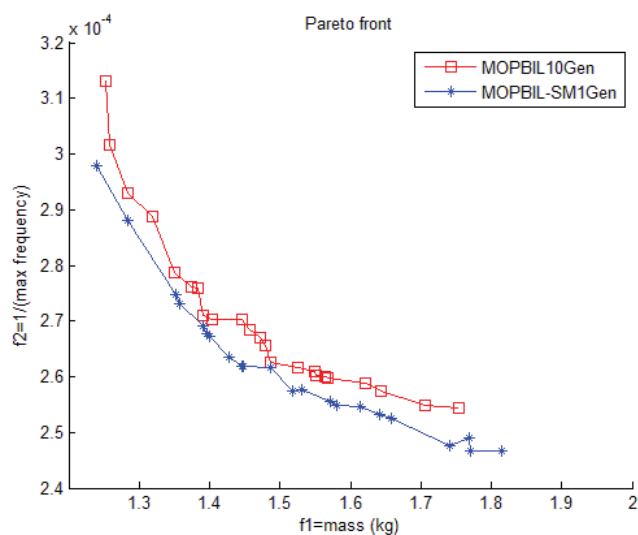
ผลการออกแบบซึ่งอยู่ในรูปของแนวพาเรโตแสดงในรูปที่ 5-21 จากรูปเป็นแนวพาเรโตที่วงวนต่างๆ ของการรัน MOPBIL-SM จากวงวนแรกจนถึงวงวนที่ 10 จากรูปจะเห็นว่าแนวพาเรโตที่วงวนที่ 10 มี

ความแตกต่างจากแนวพาเรโตของวงวนที่ 1 เล็กน้อย นั้นหมายความว่า การใช้แบบจำลองทดแทนในชั้นเริ่มต้นมีประสิทธิภาพในการหาผลเฉลยสูงมาก

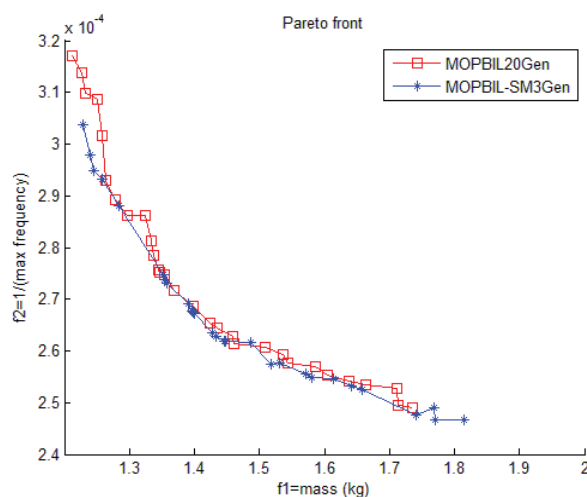


รูปที่ 5-21 แนวพาเรโตที่ได้จาก MOPBIL-SM

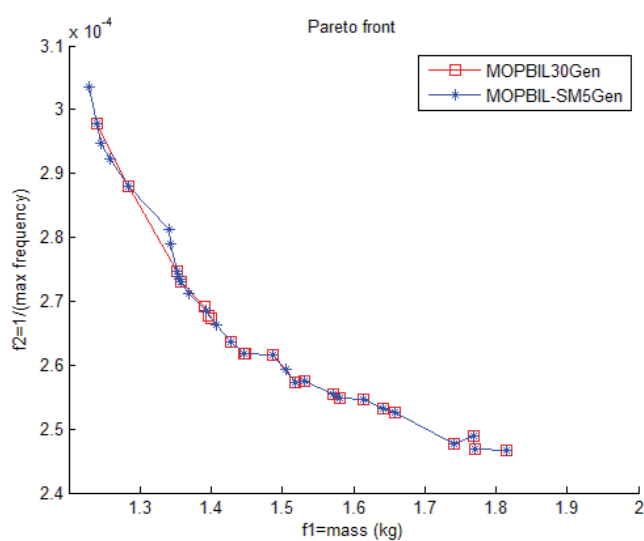
เพื่อเปรียบเทียบกับ MOPBIL ที่ไม่ใช่แบบจำลองทดแทน ทำการประยุกต์ใช้ MOPBIL หาผลเฉลยพาเรโตของปัญหาการออกแบบนี้ด้วยขนาดประชากร 30 และจำนวนวงวนเป็น 30 นั้นหมายความว่า MOPBIL แบบไม่ใช่แบบจำลองทดแทนใช้จำนวนครั้งในการวิเคราะห์ทางไฟไนต์เอลิเมนต์ทั้งหมด $30 \times 30 = 900$ ครั้ง ส่วน MOPBIL-SM ใช้ $100 + 10 \times 30 = 400$ ครั้ง รูปที่ 5-22, 5-23 และ 5-24 แสดงการเปรียบเทียบพัฒนาการของแนวพาเรโตจากการประยุกต์ใช้ MOPBIL และ MOPBIL-SM จากรูปทั้งสามพบว่า MOPBIL-SM ให้ผลที่เท่าเทียมกับ MOPBIL ที่ใช้การวิเคราะห์ไฟไนต์เอลิเมนต์ถึง 900 ครั้ง ในขณะที่ MOPBIL-SM ใช้แค่ $100 + 30 = 130$ ครั้ง ดังนั้น เราสามารถกล่าวได้ว่า แผนการออกแบบเชิงวิวัฒนาการแบบผสมที่นำเสนอในการศึกษานี้ มีประสิทธิภาพและประสิทธิผลมากสำหรับการออกแบบชิ้นส่วนเชื่อมกระปุกเกียร์



รูปที่ 5-22 เปรียบเทียบแนวพารेटอ MOPBIL 10 Generations กับ MOPBIL-SM 1 Generation

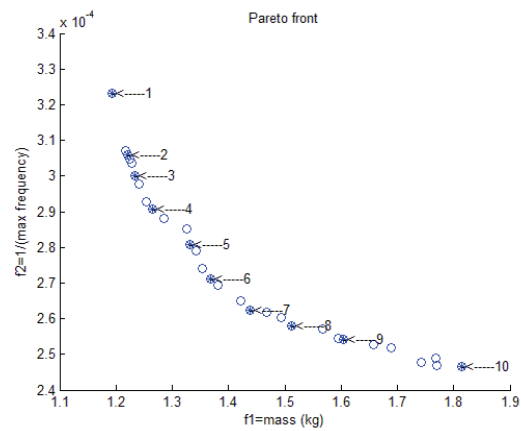


รูปที่ 5-23 เปรียบเทียบแนวพารेटอ MOPBIL 20 Generations กับ MOPBIL-SM 3 Generations

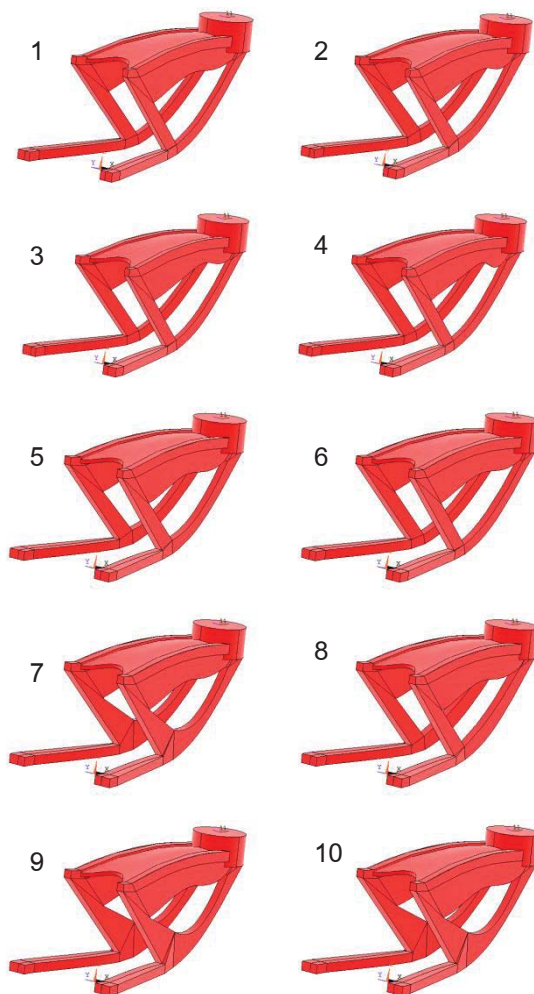


รูปที่ 5-24 เปรียบเทียบแนวพารेटอ MOPBIL 30 Generations กับ MOPBIL-SM 5 Generations

รูปที่ 5-25 แสดงแนวพาเรโตสุดท้ายที่ได้จากการประยุกต์ใช้ MOPBIL-SM ส่วนรูปทรงของชิ้นงานที่ตำแหน่งผลเฉลยต่างๆ แสดงในรูปที่ 5-26 ชิ้นงานจะมีรูปทรงคล้ายกัน แต่จะมีขนาดแตกต่างกัน และชิ้นงานทั้งหมดผ่านเงื่อนไขบังคับของการออกแบบทั้งหมด



รูปที่ 5-25 แนวพาเรโตที่ได้จาก MOPBIL-SM



รูปที่ 5-26 ชิ้นส่วนยานยนต์ที่ได้จากแนวพาเรโตรูปที่ 5-25

5.4 การออกแบบชิ้นส่วนทางกล

การออกแบบกรณีศึกษาที่สามเป็นการออกแบบหลายเป้าหมายของแขนรับโมเมนต์บิดสามมิติ โครงสร้างของแขนนี้แสดงในรูปที่ 5-27 ชิ้นส่วนทางกลรูปแบบนี้ สามารถใช้งานได้หลายประเภท ปัญหาการออกแบบที่เหมาะสมที่สุดแบบหลายเป้าหมายของชิ้นส่วนทางกลนี้ สามารถแสดงได้ดังนี้

$$\text{Min: } \mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x})] \quad (5.4)$$

Subject to

$$\sigma_{\max} \leq \sigma_{\text{all}}$$

$$0.03 \leq R_1 \leq 0.05$$

$$0.02 \leq r_1 \leq 0.05$$

$$0.01 \leq t \leq 0.025$$

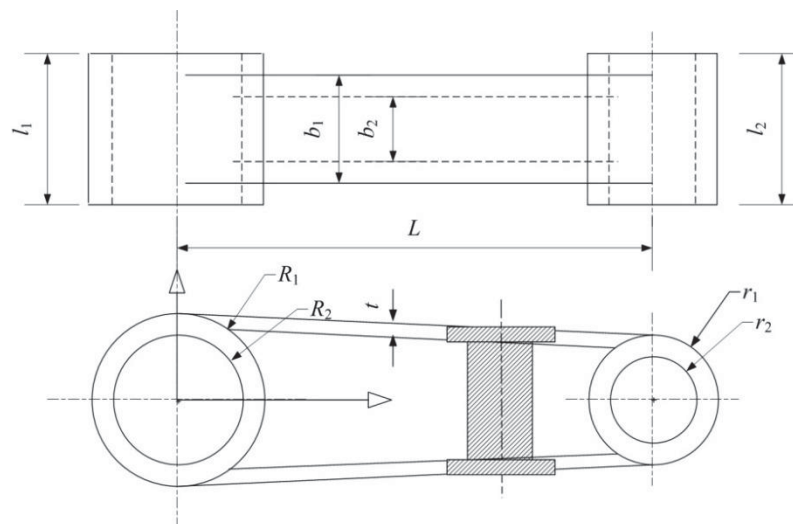
$$0.035 \leq b_1 \leq 0.075$$

$$0.015 \leq b_2 \leq 0.03$$

$$0.035 \leq l_1 \leq 0.075$$

$$0.035 \leq l_2 \leq 0.075$$

เมื่อ $\mathbf{x} = [R_1, r_1, t, b_1, b_2, l_1, l_2]^T$ คือเวกเตอร์ตัวแปรออกแบบ ในการศึกษานี้ใช้หน่วยของ SI หากไม่มีการระบุเป็นอย่างอื่น กำหนดให้ R_2 และ r_2 มีค่าเป็น 0.0225 m และ 0.015 m ตามลำดับ f_1 คือน้ำหนักของโครงสร้าง และ f_2 คือระยะยุบตัวสูงสุดที่เกิดขึ้นเมื่อโครงสร้างอยู่ภายใต้ภาระกระทำ สามรูปแบบ การหาค่า f_2 ต่ำสุดจะสมมูลกับการหาค่าความเค้นดึงสูงสุดของโครงสร้าง ค่า $\sigma_{\max}$ คือค่าความเค้น von Mises สูงสุดเนื่องจากแรงกระทำทั้งสามกรณี ส่วน σ_{all} คือค่าความเค้นสูงสุดที่ยอมให้เกิดขึ้นบนชิ้นส่วนนี้ การหาค่าฟังก์ชันออกแบบเหล่านี้ทำได้โดยการวิเคราะห์ทางไฟไนท์เอลิเมนต์ ซึ่งจะใช้เวลาในการคำนวณสูงมากเมื่อค่าของ b_1 มีค่าเข้าใกล้ค่าของ l_1 และ l_2 เพราะจะส่งผลให้ต้องใช้เอลิเมนต์เป็นจำนวนมากในการสร้างแบบจำลองไฟไนท์เอลิเมนต์



รูปที่ 5-27 แขนรับโมเมนต์บิดสามมิติ

วิธีวิวัฒนาการแบบผสมสำหรับการศึกษาในหัวข้อนี้คือวิธี PBIL เช่นเดียวกับหัวข้อที่ผ่านมา อย่างไรก็ตามในงานนี้ได้ทำการศึกษาเปรียบเทียบขั้นตอนวิธีวิวัฒนาการ PBIL แบบผสม สองรูปแบบ กล่าวคือ

- Strategy1: สำหรับแผนการหาค่าเหมาะที่สุดนี้ MOPBIL ถูกใช้ในการค้นหาผลเฉลยเหมาะที่สุดของพารามิเตอร์ประมาณ 2-3 วงวน จากนั้นนำประชากรในวงวนที่ผ่านมาทั้งหมดมาสร้างแบบจำลองทดแทน จากนั้นทำการประยุกต์ใช้ PBIL หาผลเฉลยของพารามิเตอร์ภายใต้การหาค่าฟังก์ชันด้วยแบบจำลองทดแทน เมื่อเสร็จสิ้นการหาผลเฉลยด้วยแบบจำลองทดแทนแล้ว เลือกผลเฉลยที่ไม่ถูกรอบงำจำนวนหนึ่งเพื่อมาหาค่าจริงของผลเฉลยเหล่านั้น และกลุ่มของผลเฉลยที่ได้ในขั้นตอนนี้เป็นผลเฉลยตั้งต้นของวิธี PBIL ซึ่งจะถูกดำเนินการต่ออีกระยะเวลาหนึ่งเพื่อพัฒนาผลเฉลยที่ไม่ถูกรอบงำให้ดีขึ้นไปอีก

- Strategy2: มีรูปแบบการดำเนินการคล้ายกับ Strategy 1 โดยมีข้อแตกต่างคือ แผนการนี้ใช้ LHS เพื่อกำหนดกลุ่มของผลเฉลยเพื่อใช้ในการสร้างแบบจำลองทดแทน

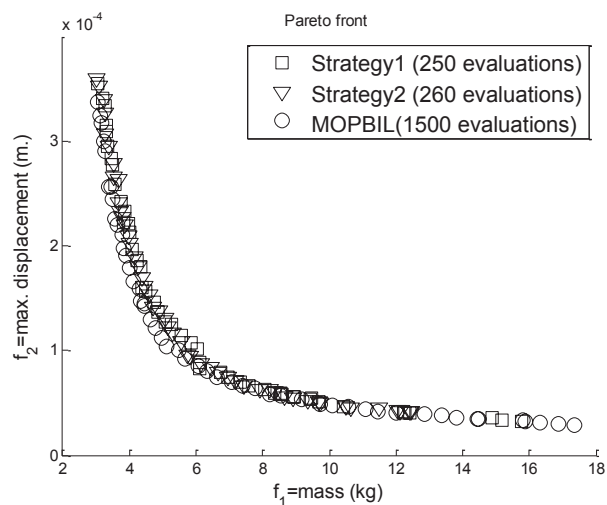
เงื่อนไขในการรันขั้นตอนวิธีวิวัฒนาการเหล่านี้คือ สำหรับ Strategy1 ใช้ MOPBIL หาผลเฉลย 3 วงวน ด้วยขนาดของประชากรเป็น 30 เพื่อหาจุดสำหรับสร้างแบบจำลองทดแทน จากนั้นรัน MOPBIL เพื่อหาผลเฉลยจากการใช้แบบจำลองทดแทน หลังจากนั้นรับ MOPBIL เพื่อปรับปรุงแนวพารามิเตอร์ที่ได้ต่ออีก 5 วงวน สรุปแล้วใช้การวิเคราะห์ทางไฟไนต์เอลิเมนต์ทั้งสิ้น $30 \times 3 + 30 \times 6 = 270$ ครั้ง สำหรับ Strategy2 วิธี LHS สร้างผลเฉลยขึ้นมา 100 จุด จากนั้นสร้างแบบจำลองทดแทน และรัน MOPBIL เพื่อหาผลเฉลยจากแบบจำลองทดแทนนั้น เมื่อได้คำตอบตั้งต้นจากการรัน MOPBIL กับแบบจำลองทดแทนแล้ว รัน MOPBIL ต่ออีก 10 วงวนเพื่อปรับปรุงแนวของพารามิเตอร์ที่ได้ สรุปคือ Strategy2 ใช้จำนวนครั้งในการวิเคราะห์ไฟไนต์เอลิเมนต์ทั้งหมด 260 ครั้ง เพื่อวัดสมรรถนะของขั้นตอนวิธีวิวัฒนาการแบบผสมทั้งสองแผนการ MOPBIL ที่ไม่ใช่แบบจำลองทดแทน จะถูกใช้ในการค้นหาคำตอบของปัญหาการออกแบบนี้ โดยใช้ขนาดของประชากร 30 และมีจำนวนวงวนเป็น 50 หรือกล่าวอีกนัยหนึ่งคือ ใช้การวิเคราะห์ทางไฟไนต์เอลิเมนต์ทั้งสิ้น 1500 ครั้ง

ตารางที่ 5-7 แสดงการตรวจสอบความแม่นยำของแบบจำลองฟังก์ชันฐานหลักแนวกว้างที่สร้างผลเฉลยด้วย Strategy1 และ Strategy2 จากตารางพบว่า ยกเว้นในกรณีของฟังก์ชันเป้าหมายตัวแรกคือน้ำหนักของโครงสร้าง แบบจำลองทดแทนของ Strategy1 มีความแม่นยำสูงกว่าแบบจำลองทดแทนที่สร้างจาก Strategy 2

ตารางที่ 5-7 RMSE ของค่าฟังก์ชันเงื่อนไขบังคับและฟังก์ชันเป้าหมาย g_i คือเงื่อนไขบังคับสำหรับภาระกรณี i -th

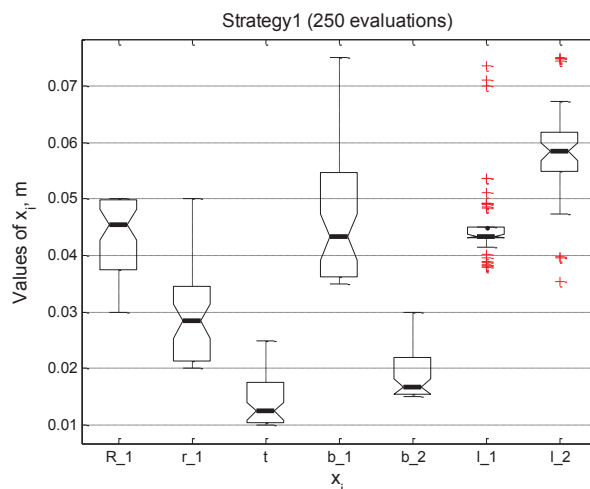
Methods	f_1	f_2	g_1	g_2	g_3
Strategy1	0.5193	0.6733×10^{-4}	6.6089×10^7	4.9847×10^7	5.4952×10^7
Strategy2	0.4902	0.7026×10^{-4}	8.4202×10^7	8.4089×10^7	9.3479×10^7

แนวพาเรโตของปัญหาการออกแบบที่ได้จากการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการทั้งสาม แสดงในรูปที่ 5-28 ไฮเปอร์วอลุ่มของแนวพาเรโตที่ได้จาก Strategy1, Strategy2, และ MOPBIL แบบไม่มีแบบจำลองทดแทนมีค่าเป็น 0.0027, 0.0024, และ 0,0025 ตามลำดับ ทั้งนี้ จุดอ้างอิงที่ใช้ในการคำนวณค่า HV คือ $[f_1 = 17 \text{ kg}, f_2 = 0.00025 \text{ m}]$ จากผลนี้ แสดงว่า แผนการสร้างแบบจำลองทดแทนด้วย Strategy1 มีประสิทธิภาพสูงกว่าการรันสองแบบที่เหลือ สาเหตุก็มาจากการที่เราใช้ MOPBIL ค้นหาผลเฉลยเพื่อนำมาสร้างแบบจำลองทดแทนนั้น ทำให้สามารถหาผลเฉลยที่เป็นไปได้ (feasible solution) ได้ดีกว่าการใช้วิธี LHS

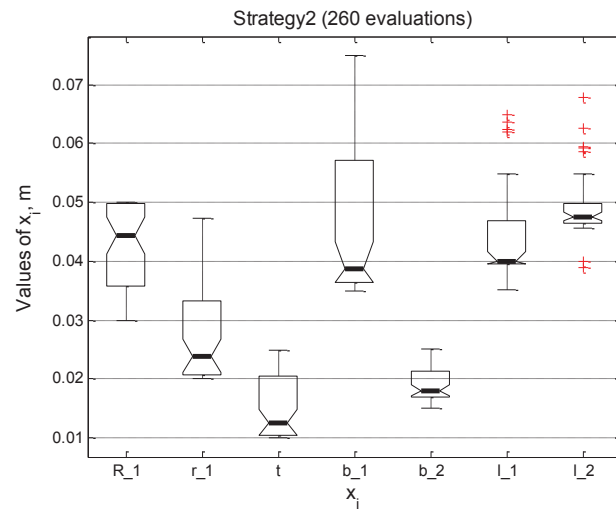


รูปที่ 5-28 แนวพาเรโต จากขั้นตอนวิธีวิวัฒนาการต่างๆ

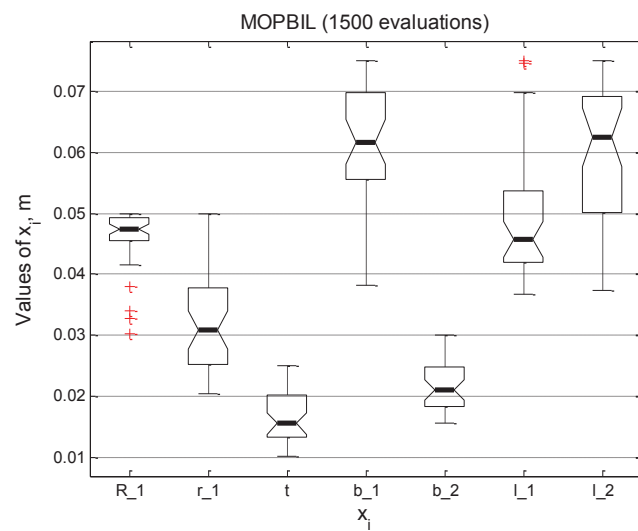
การกระจายตัวของตัวแปรออกแบบของผลเฉลยที่ไม่ถูกครอบงำที่หาได้จากการใช้ Strategy1, Strategy2, และ MOPBIL แสดงในรูปที่ 5-29, 5-30 และ 5-31 ตามลำดับ จากรูปทั้งสามอาจกล่าวได้ว่าการกระจายตัวของตัวแปรออกแบบที่เป็นสมาชิกของแนวพาเรโตทั้งสามในรูปที่ 5-29 มีรูปแบบใกล้เคียงกัน



รูปที่ 5-29 การกระจายตัวของตัวแปรออกแบบจาก Strategy1



รูปที่ 5-30 การกระจายตัวของตัวแปรออกแบบจาก Strategy2



รูปที่ 5-31 การกระจายตัวของตัวแปรออกแบบจาก MOPBIL

การออกแบบอุปกรณ์ระบายความร้อนแบบครีบนํ้า

6.1 ความนำ

คอมพิวเตอร์เข้ามามีบทบาทสำคัญต่อระบบอุตสาหกรรมและชีวิตประจำวันของเราเป็นอย่างมากในปัจจุบัน และส่วนประกอบที่สำคัญที่สุดของคอมพิวเตอร์ก็คือหน่วยประมวลผลกลาง (CPU) หรือเรียกอีกอย่างหนึ่งว่าไมโครโปรเซสเซอร์ ซึ่งเปรียบเสมือนมันสมองของคอมพิวเตอร์ ทำหน้าที่ประมวลผล เก็บคำสั่ง และข้อมูลที่ได้รับ ตลอดจนควบคุมระบบการทำงานของคอมพิวเตอร์ วัดความสามารถในการประมวลผลของระบบคอมพิวเตอร์จะขึ้นอยู่กับความเร็วของหน่วยประมวลผลกลางเป็นอันดับแรก ซึ่งหากเลือกใช้หน่วยประมวลผลกลางที่มีความเร็วในการทำงานสูงจะทำให้เครื่องคอมพิวเตอร์มีความเร็วในการทำงานโดยรวมสูงและมีความถูกต้องแม่นยำสูงตามไปด้วย

ในปัจจุบันอุปกรณ์อิเล็กทรอนิกส์มีขนาดเล็กลงมาก ยกตัวอย่างเช่น คอมพิวเตอร์พกพา ซึ่งเน้นให้มีขนาดเล็ก กะทัดรัด พกพาได้สะดวก แต่จะต้องมีประสิทธิภาพในการประมวลผลที่สูงขึ้น การออกแบบดังกล่าวส่งผลให้ความหนาแน่นทางความร้อนภายในอุปกรณ์อิเล็กทรอนิกส์มีมากขึ้น โดยเฉพาะในหน่วยประมวลผลกลาง ความร้อนหนาแน่นจะสะสมเพิ่มขึ้นเรื่อย ๆ หากไม่มีการออกแบบระบบระบายความร้อนที่ดีพอ อาจทำให้เกิดความเสียหายแก่อุปกรณ์อิเล็กทรอนิกส์ เกิดการไหม้ของหน่วยประมวลผลกลางหรือทำให้การวิเคราะห์ข้อมูลของอุปกรณ์มีความคลาดเคลื่อน และที่สำคัญที่สุดคือทำให้อายุการใช้งานของอุปกรณ์อิเล็กทรอนิกส์นั้น ๆ สั้นลงไปด้วย จากที่กล่าวมาข้างต้นนับเป็นความท้าทายของวิศวกรด้านความร้อนในยุคปัจจุบันที่จะต้องออกแบบระบบหรืออุปกรณ์ระบายความร้อนเพื่อรองรับแนวโน้มการเจริญเติบโตทางด้านไอที

การจัดการปัญหาด้านความร้อนของอุปกรณ์อิเล็กทรอนิกส์ในปัจจุบันจะใช้อุปกรณ์ระบายความร้อน (Heat Sinks) มาทำหน้าที่ระบายความร้อนออกจากชิป (Chip) หรือหน่วยประมวลผลกลางของคอมพิวเตอร์ ซึ่งอุปกรณ์ระบายความร้อนที่นิยมใช้งานในปัจจุบันสามารถแยกประเภทตามตัวกลางในการนำความร้อนออกจากอุปกรณ์อิเล็กทรอนิกส์ได้ 3 ประเภทหลัก ๆ ดังนี้

6.1.1 อุปกรณ์ระบายความร้อนด้วยอากาศ

อุปกรณ์ระบายความร้อนที่ใช้อากาศในการระบายความร้อน (Air-Cooled Heat Sinks) เป็นอุปกรณ์ระบายความร้อนที่นิยมกันอย่างแพร่หลาย ด้วยการใช้งานที่ง่าย ไม่ยุ่งยากในการติดตั้งและการบำรุงรักษา โดยประสิทธิภาพที่ออกมาขึ้นขึ้นอยู่กับความสามารถของผู้ออกแบบ เทคนิคและ เทคโนโลยีที่นำมาใช้ในการออกแบบ หลักการระบายความร้อนด้วยอากาศ ใช้หลักการของการถ่ายเทความร้อน 3 รูปแบบ คือ การนำความร้อน (Conduction) การพาความร้อน (Convection) และการแผ่รังสีความร้อน (Radiation)

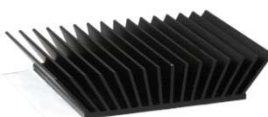
สำหรับอุปกรณ์ระบายความร้อนด้วยอากาศที่นิยมใช้ระบายความร้อนออกจากอุปกรณ์อิเล็กทรอนิกส์มีอยู่ 2 ชนิดหลักดังนี้

- อุปกรณ์ระบายความร้อนแบบครีบบแผ่น (Plate-Fin Heat sinks)

อุปกรณ์ระบายความร้อนแบบครีบบแผ่นเป็นอุปกรณ์ระบายความร้อนด้วยอากาศที่นิยมใช้กันมากที่สุด เนื่องจากง่ายในการผลิต และราคาถูก ทั้งนี้อุปกรณ์ระบายความร้อนแบบครีบบแผ่นมีทั้งการออกแบบให้มีครีบบางสี่เหลี่ยม ครีบบางเหลี่ยมหรือแม้แต่แผ่นครีบบางที่มีลักษณะเอียงเพื่อเป็นการเพิ่มพื้นที่การส่งผ่านความร้อนจากอุปกรณ์ระบายความร้อนสู่อากาศ ตัวอย่างอุปกรณ์ระบายความร้อนแบบครีบบางสี่เหลี่ยมตรงแสดงไว้ดังรูปที่ 6-1 และครีบบางเอียงแสดงไว้ในรูปที่ 6-2



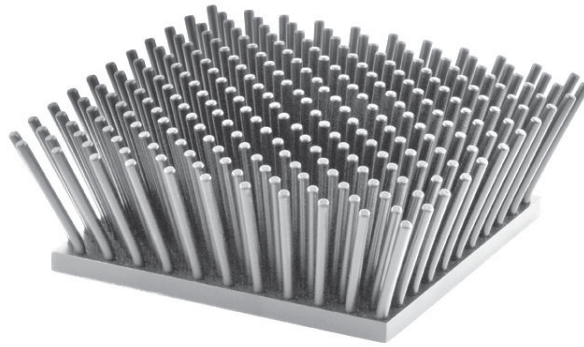
รูปที่ 6-1 อุปกรณ์ระบายความร้อนแบบครีบบางสี่เหลี่ยมตรง



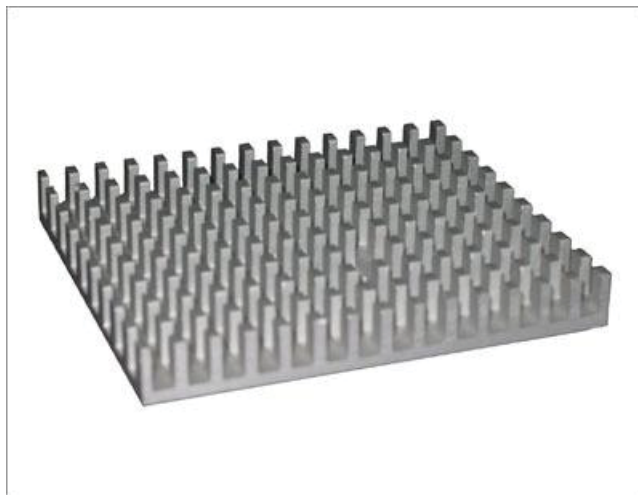
รูปที่ 6-2 อุปกรณ์ระบายความร้อนแบบครีบบางเอียง

- อุปกรณ์ระบายความร้อนแบบครีบบาง (Pin-Fin Heat sinks)

อุปกรณ์ระบายความร้อนแบบครีบบางเป็นอุปกรณ์ระบายความร้อนด้วยอากาศที่ออกแบบมาเพื่อเพิ่มพื้นที่การส่งผ่านความร้อนให้มากขึ้น ซึ่งอุปกรณ์ระบายความร้อนแบบนี้มีข้อดีก็คือการไหลของอากาศที่ผ่านครีบบางสามารถกระจายไปทุกทิศทางรอบครีบบางระบายความร้อน อุปกรณ์ระบายความร้อนแบบครีบบางที่มีใช้ในปัจจุบันมีทั้งครีบบางที่มีหน้าตัดสี่เหลี่ยม หน้าตัดวงกลม หน้าตัดวงรี และหน้าตัดอื่น ๆ เพื่อเพิ่มพื้นที่ในการส่งผ่านความร้อนจากอุปกรณ์ระบายความร้อนสู่อากาศ ตัวอย่างของอุปกรณ์ระบายความร้อนแบบครีบบางหน้าตัดวงกลมแสดงไว้ดังรูปที่ 6-3 และหน้าตัดสี่เหลี่ยมแสดงไว้ในรูปที่ 6-4



รูปที่ 6-3 อุปกรณ์ระบายความร้อนแบบครีบท่งที่มีครีบน้้าัดดวงกลม



รูปที่ 6-4 อุปกรณ์ระบายความร้อนแบบครีบท่งที่มีครีบน้้าัดสี่เหลี่ยม

สำหรับตัวอุปกรณ์ระบายความร้อนด้วยอากาศมีส่วนประกอบหลัก 3 ส่วนซึ่งประกอบด้วยส่วนต่าง ๆ ดังต่อไปนี้

1) ส่วนฐานของอุปกรณ์ระบายความร้อน (Heat Sink Base)

ทำหน้าที่เป็นตัวรับความร้อนหรือถ่ายเทความร้อนจากแหล่งกำเนิดความร้อน (Chip/CPU) เพื่อส่งผ่านไปยังส่วนที่ทำหน้าที่ถ่ายเทความร้อนสู่บรรยากาศอีกต่อหนึ่ง ซึ่งส่วนฐานนี้เป็นส่วนที่สัมผัสกับบริเวณที่ต้องการถ่ายเทความร้อนออกโดยตรง นั่นก็คือหน่วยประมวลผลกลางของคอมพิวเตอร์หรือตัวชิปอื่น ๆ นั่นเอง

2) ครีกระบายความร้อน (Fin)

ครีบท่งเป็นส่วนที่ทำหน้าที่ถ่ายเทความร้อนที่ฐานของอุปกรณ์ระบายความร้อนรับมาจากตัวอุปกรณ์อิเล็กทรอนิกส์หรือบริเวณที่ต้องการระบายความร้อนออกไปสู่บรรยากาศ เพื่อให้มีอุณหภูมิลดลง ซึ่งในการระบายความร้อนเพื่อให้มีประสิทธิภาพดีขึ้นอาจจะต้องมีการใช้พัดลมเข้ามาช่วยในการเป่าอากาศ

3) พัดลมเป่าอากาศ (Fan)

พัฒมเป็นอุปกรณ์ที่ทำหน้าที่เพิ่มประสิทธิภาพให้กับการระบายความร้อนของอุปกรณ์ระบายความร้อน โดยทำหน้าที่เป่าอากาศให้มีความเร็วเพิ่มขึ้น เพื่อให้อากาศสามารถพาความร้อนออกจากอุปกรณ์ระบายความร้อนได้ดีขึ้น

6.1.2 อุปกรณ์ระบายความร้อนด้วยของเหลว

อุปกรณ์ระบายความร้อนด้วยของเหลว (Liquid-Cooled Heat Sinks) มีหลักการทำงานเช่นเดียวกับอุปกรณ์ระบายความร้อนด้วยอากาศ หากแต่เปลี่ยนสารตัวกลางที่ทำหน้าที่นำพาความร้อนออกจากแหล่งหรือพื้นที่ที่ต้องการระบายความร้อนจากเดิมที่เป็นอากาศมาเป็นของเหลวแทน รูปทรงและองค์ประกอบของตัวอุปกรณ์ระบายความร้อนด้วยของเหลวนั้นจะแตกต่างจากอุปกรณ์ระบายความร้อนด้วยอากาศอย่างสิ้นเชิง จุดแตกต่างจุดแรกที่เราเห็นได้เด่นชัดที่สุดคือ อุปกรณ์ระบายความร้อนด้วยของเหลวประกอบด้วยองค์ประกอบที่ทำงานแยกเป็นส่วนกันอย่างอิสระ ถึงแม้ในบางครั้งอาจจะถูกติดตั้งรวมกันในพื้นที่เดียวกันก็ตาม ซึ่งแต่ละองค์ประกอบของอุปกรณ์ระบายความร้อนด้วยของเหลวจะมีหน้าที่ของตัวเอง หากขาดส่วนหนึ่งส่วนใดไปก็ จะทำให้ไม่สามารถทำงานได้ ตัวอย่างอุปกรณ์ระบายความร้อนด้วยของเหลวแสดงไว้ในรูปที่ 6-5

สำหรับส่วนประกอบหลักของอุปกรณ์ระบายความร้อนด้วยของเหลว ประกอบด้วยองค์ประกอบดังต่อไปนี้



รูปที่ 6-5 อุปกรณ์ระบายความร้อนด้วยของเหลว

1) อุปกรณ์แลกเปลี่ยนความร้อน (Water Block)

อุปกรณ์แลกเปลี่ยนความร้อนทำหน้าที่รับความร้อนจากหน่วยประมวลผลกลางหรือพื้นที่ที่ต้องการถ่ายเทความร้อนและถ่ายเทความร้อนดังกล่าวให้กับสารตัวกลางเพื่อนำความร้อนออกไปทิ้ง ซึ่งลักษณะของอุปกรณ์แลกเปลี่ยนความร้อนนี้จะมีลักษณะที่แตกต่างกันออกไปตามการออกแบบของผู้ผลิต

2) อุปกรณ์ถ่ายเทความร้อนออกจากของเหลว (Radiator)

ทำหน้าที่นำความร้อนที่ของเหลวหรือสารตัวกลางรับมาจากอุปกรณ์แลกเปลี่ยนความร้อนออกจากของเหลว เพื่อให้ของเหลวมีอุณหภูมิลดลง

3) อุปกรณ์ที่ทำให้สารตัวกลางเคลื่อนที่ (Water Pump)

เป็นอุปกรณ์ที่สำคัญที่สุดของอุปกรณ์ระบายความร้อนด้วยของเหลว เนื่องจากเป็นตัวที่ทำหน้าที่ผลักดันให้สารตัวกลางไหลเวียนเพื่อแลกเปลี่ยนความร้อนและทิ้งความร้อนระหว่างอุปกรณ์แลกเปลี่ยนความร้อนกับอุปกรณ์ถ่ายเทความร้อนออกจากของเหลว

ประสิทธิภาพการถ่ายเทความร้อนออกจากอุปกรณ์อิเล็กทรอนิกส์ของอุปกรณ์ระบายความร้อนด้วยของเหลวมีประสิทธิภาพสูงเมื่อเทียบกับอุปกรณ์ระบายความร้อนด้วยอากาศเนื่องจากค่าความร้อนจำเพาะของของเหลวมีค่ามากกว่าความร้อนจำเพาะของอากาศ

6.1.3 อุปกรณ์ระบายความร้อนที่ใช้การเปลี่ยนสถานะของตัวกลางในการถ่ายเทความร้อน

อุปกรณ์ระบายความร้อนที่ใช้การเปลี่ยนสถานะของตัวกลางในการถ่ายเทความร้อนหรือฮีทไปป์ (Heat Pipe) สามารถนำความร้อนจากแหล่งจ่ายความร้อนไปยังแหล่งทิ้งความร้อนได้อย่างมีประสิทธิภาพการทำงานของฮีทไปป์เป็นการทำงานแบบไม่มีส่วนใดเคลื่อนไหว โครงสร้างพื้นฐานของฮีทไปป์เป็นท่อทองแดงปิดหัวและท้าย ภายในบรรจุด้วยสารทำความเย็น เช่น ฟรอน แอมโมเนีย ออกซิเจน มีเทน และน้ำ สารที่บรรจุอยู่ภายในท่อเมื่อได้รับความร้อนจะเกิดการระเหยกลายเป็นไอและเคลื่อนไปสู่ปลายอีกด้านหนึ่งในระหว่างการเคลื่อนตัวสารตัวกลางจะคายความร้อนและกลายเป็นของเหลวที่บริเวณผิวภายในของท่อและไหลกลับไปสู่ปลายท่อที่รับความร้อนอีกครั้งหนึ่ง เป็นการวนเวียนแลกเปลี่ยนความร้อนและถ่ายเทความร้อนทิ้ง

ในปัจจุบันมีการนำเอาฮีทไปป์มาใช้เพื่อช่วยระบายความร้อนในเครื่องคอมพิวเตอร์ส่วนบุคคลและคอมพิวเตอร์แบบพกพารุ่นใหม่ ๆ ที่มีความสามารถในการประมวลผลสูง เนื่องจากเครื่องมืออิเล็กทรอนิกส์เหล่านี้ใช้ไอชิรุ่นใหม่ที่กำลังวัตต์สูงขึ้น มีการนำไปใช้ระบายความร้อนในแผงควบคุม แผงวงจร เครื่องเสียง เพื่อเพิ่มประสิทธิภาพในการระบายความร้อน และทำให้มีขนาดและน้ำหนักลดลง นอกจากนี้ยังมีการประยุกต์ใช้ฮีทไปป์ร่วมกับอุปกรณ์ระบายความร้อนด้วยอากาศ โดยติดตั้งฮีทไปป์ไว้ที่ฐานของอุปกรณ์แลกเปลี่ยนความร้อนด้วยอากาศ เพื่อให้ระบายความร้อนออกจากฐานไปสู่ครีระบายความร้อนได้มีประสิทธิภาพมากขึ้น ตัวอย่างของฮีทไปป์แสดงไว้ในรูปที่ 6-6



รูปที่ 6-6 อุปกรณ์ระบายความร้อนที่ใช้การเปลี่ยนสถานะของตัวกลางหรือฮีทไปป์

จากอุปกรณ์ระบายความร้อนทั้ง 3 ประเภทที่กล่าวมาข้างต้น เมื่อทำการศึกษาข้อมูลเปรียบเทียบข้อดีและข้อเสียของอุปกรณ์ระบายความร้อนดังกล่าว สามารถสรุปออกมาเป็นข้อ ๆ ได้ดังต่อไปนี้

1) เปรียบเทียบด้านราคา

จากอุปกรณ์ระบายความร้อนทั้ง 3 ประเภท ถึงแม้ว่าอุปกรณ์ระบายความร้อนด้วยของเหลวและการระบายความร้อนด้วยฮีทปั๊มจะมีประสิทธิภาพในการระบายความร้อนดีกว่าอุปกรณ์ระบายความร้อนด้วยอากาศ แต่ในปัจจุบันราคาของอุปกรณ์ระบายความร้อนทั้งสองประเภทยังสูงมากเมื่อเปรียบเทียบกับอุปกรณ์ระบายความร้อนด้วยอากาศ

2) เปรียบเทียบด้านการบำรุงรักษา

จากข้อมูลส่วนประกอบของอุปกรณ์ระบายความร้อนในแต่ละประเภทที่กล่าวมาพบว่าอุปกรณ์ระบายความร้อนด้วยของเหลวมีความยากลำบากในการบำรุงรักษาที่สุด เนื่องจากมีอุปกรณ์หลายตัวซึ่งต้องทำงานประสานกัน ในขณะที่อุปกรณ์ระบายความร้อนด้วยอากาศง่ายต่อการบำรุงรักษามากที่สุดเนื่องจากตัวอุปกรณ์มีแค่พัดลมเป่าอากาศและครีบบระบายความร้อนเท่านั้น

3) เปรียบเทียบข้อมูลด้านการติดตั้ง

การติดตั้งอุปกรณ์ระบายความร้อนด้วยของเหลวมีความยากลำบากที่สุด เนื่องจากจะต้องคำนึงถึงอันตรายที่อาจจะเกิดต่ออุปกรณ์อิเล็กทรอนิกส์อันเนื่องมาจากการรั่วไหลของของเหลวที่ใช้เป็นตัวกลางในการระบายความร้อน ฮีทปั๊มก็ต้องคำนึงถึงการรั่วไหลเช่นเดียวกัน แต่อุปกรณ์ระบายความร้อนด้วยอากาศไม่ต้องคำนึงถึงอันตรายจากเหตุผลในข้อนี้

จากข้อดีและข้อเสียทั้งสามข้อข้างต้นทำให้อุปกรณ์ระบายความร้อนด้วยอากาศได้รับความนิยมและใช้กันอย่างแพร่หลายที่สุดในปัจจุบัน แต่การระบายความร้อนด้วยอากาศมีข้อเสียคือระบายความร้อนออกจากอุปกรณ์อิเล็กทรอนิกส์ได้น้อยเนื่องจากค่าความร้อนจำเพาะของอากาศมีค่าต่ำกว่าน้ำหรือของเหลวชนิดอื่น ถ้าจะให้ระบายความร้อนได้ดีขึ้นจะต้องใช้อุปกรณ์ระบายความร้อนที่มีพื้นที่ในการส่งผ่านความร้อนมากขึ้น นั่นคือจะต้องมีขนาดใหญ่และใช้พื้นที่ในการติดตั้งมากขึ้น ซึ่งสวนกระแสกับความนิยมอุปกรณ์อิเล็กทรอนิกส์ในปัจจุบันที่ต้องมีขนาดเล็ก บาง เบา กะทัดรัด เหมาะแก่การพกพา แต่มีประสิทธิภาพในการประมวลผลสูงหรืออีกหนึ่งวิธีที่สามารถทำให้ประสิทธิภาพการระบายความร้อนสูงขึ้นได้นั้นก็คือการเพิ่มความเร็วของอากาศที่ไหลเข้าไประบายความร้อน แต่สิ่งที่ตามมาคือกำลังไฟที่จะต้องใช้ในการจ่ายให้แก่พัดลมเป่าอากาศจะมีค่ามากขึ้น และเกิดเสียงดัง

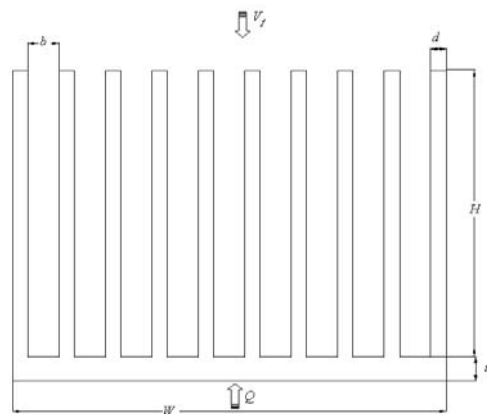
ดังนั้นการศึกษานี้จึงทำการศึกษาเพื่อออกแบบอุปกรณ์ระบายความร้อนด้วยอากาศแบบครีบบแผ่นซึ่งเป็นอุปกรณ์ระบายความร้อนที่นิยมใช้กันมากที่สุด โดยใช้การออกแบบด้วยการหาค่าเหมาะที่สุดเพื่อให้ได้อุปกรณ์ระบายความร้อนที่เหมาะสมกับกระแสความนิยมอุปกรณ์อิเล็กทรอนิกส์ในปัจจุบัน สำหรับการวิเคราะห์การถ่ายเทความร้อนของอุปกรณ์ระบายความร้อนด้วยอากาศแบบครีบบแผ่นจะได้กล่าวถึงในหัวข้อ 6.2 ต่อไป

6.2 สมการทางคณิตศาสตร์ของอุปกรณ์ระบายความร้อนด้วยอากาศแบบครีปแผ่น

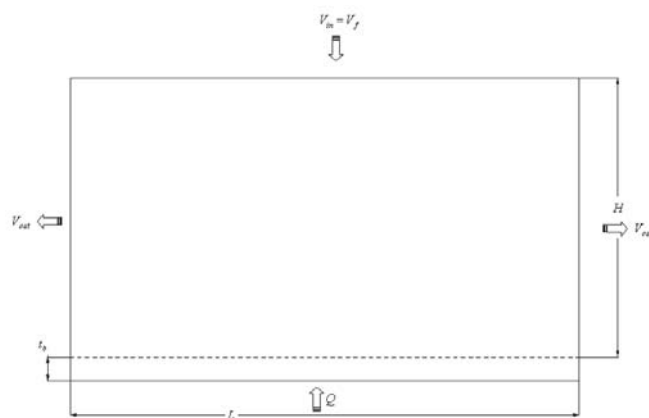
สมรรถนะทางความร้อนของอุปกรณ์ระบายความร้อนแบบครีปแผ่นขึ้นอยู่กับขนาดของพื้นที่ผิวในการส่งผ่านความร้อนจากอุปกรณ์ระบายความร้อนออกไปสู่อากาศ ซึ่งหมายความว่าในการออกแบบอุปกรณ์ระบายความร้อนจะต้องคำนึงถึงตัวแปรรูปร่างที่เหมาะสมเพื่อให้อุปกรณ์ระบายความร้อนสามารถระบายความร้อนได้ตามที่ต้องการ หากวิเคราะห์การถ่ายเทความร้อนอุปกรณ์ระบายความร้อนจะพบว่าการถ่ายเทความร้อนสามแบบด้วยกันดังนี้

- 1) การนำความร้อน เกิดขึ้นที่ฐานและครีปของอุปกรณ์ระบายความร้อน
- 2) การพาความร้อน เกิดขึ้นบริเวณพื้นที่ผิวสัมผัสระหว่างอากาศกับอุปกรณ์ระบายความร้อน
- 3) การแผ่รังสีความร้อน เกิดขึ้นเนื่องจากอุณหภูมิที่แตกต่างกันระหว่างตัวอุปกรณ์ระบายความร้อนและสิ่งแวดล้อมหรืออากาศ

สำหรับการวิเคราะห์หาค่าการถ่ายเทความร้อนของอุปกรณ์ระบายความร้อนในการศึกษาครั้งนี้ พิจารณาการถ่ายเทความร้อนเพียงสองแบบเท่านั้นคือการนำความร้อนและการพาความร้อน โดยพิจารณาพื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนแบบครีปแผ่นที่แสดงไว้ในรูปที่ 6-7 และรูปแบบการไหลของอากาศผ่านอุปกรณ์ระบายความร้อนในรูปที่ 6-8



รูปที่ 6-7 พื้นที่หน้าตัดและตัวแปรที่เกี่ยวข้องของอุปกรณ์ระบายความร้อนแบบครีปแผ่น



รูปที่ 6-8 รูปแบบการไหลของอากาศผ่านอุปกรณ์ระบายความร้อน

จากรูปที่ 6-7 และรูปที่ 6-8 สามารถวิเคราะห์เพื่อหาสมการทางคณิตศาสตร์ของการถ่ายเทความร้อนของอุปกรณ์ระบายความร้อนแบบครีบบ่น โดยพิจารณาค่าการถ่ายเทความร้อนของอุปกรณ์ระบายความร้อนตามสมการที่ (6.1) ดังนี้

$$Q = \frac{\Delta T}{R_{HS}} \quad (6.1)$$

โดยที่ Q คือ ค่าความร้อนที่ต้องการระบายออกจากอุปกรณ์อิเล็กทรอนิกส์
 ΔT คือ ผลต่างระหว่างอุณหภูมิของจุดที่ต้องการระบายความร้อนกับอุณหภูมิอ้างอิง
 R_{HS} คือ ความต้านทานความร้อนของอุปกรณ์ระบายความร้อน

สำหรับการศึกษาการออกแบบอุปกรณ์ระบายความร้อนในครั้งนี้ค่าผลต่างอุณหภูมิ ΔT เป็นไปตามสมการที่ (6.2) ดังนี้

$$\Delta T = T_{jc} - T_a \quad (6.2)$$

โดยที่ T_{jc} คือ อุณหภูมิจุดต่อระหว่างอุปกรณ์ระบายความร้อนกับอุปกรณ์อิเล็กทรอนิกส์
 T_a คือ อุณหภูมิของอากาศที่ไหลเข้ามาระบายความร้อน

ในการออกแบบอุปกรณ์ระบายความร้อนสิ่งที่ต้องคำนึงถึงเป็นอันดับแรกก็คือสมรรถนะทางความร้อนของอุปกรณ์ที่ต้องมีค่าสูง ซึ่งตัวแปรที่บ่งชี้ถึงสมรรถนะของอุปกรณ์ระบายความร้อนก็คืออุณหภูมิจุดต่อนั่นเอง ดังนั้นจากสมการที่ (6.1) และสมการที่ (6.2) เราสามารถหาค่าอุณหภูมิจุดต่อระหว่างอุปกรณ์ระบายความร้อนและอุปกรณ์อิเล็กทรอนิกส์ได้ตามสมการที่ (6.3)

$$T_{jc} = T_a + QR_{HS} \quad (6.3)$$

สำหรับการถ่ายเทความร้อนผ่านอุปกรณ์ระบายความร้อนประกอบด้วยการถ่ายเทความร้อนผ่านสามส่วนหลัก คือ การถ่ายเทความร้อนโดยการนำความร้อนผ่านฐานอุปกรณ์ระบายความร้อน การถ่ายเทความร้อนโดยการนำและการพาความร้อนผ่านครีบบ่นระบายความร้อน และการถ่ายเทความร้อนโดยการพาความร้อนผ่านพื้นที่บริเวณช่องว่างระหว่างครีบบ่น ดังนั้นเราสามารถหาค่าความต้านทานความร้อนของแต่ละส่วน เพื่อนำไปหาค่าความต้านทานความร้อนรวมของอุปกรณ์ระบายความร้อนได้ดังนี้

จากการวิเคราะห์การนำและการพาความร้อนผ่านครีบบ่นระบายความร้อน ได้สมการของความต้านทานความร้อนของครีบบ่นระบายความร้อนตามสมการที่ (6.4)

$$R_{fin} = \frac{1}{\sqrt{hPkA_c \tanh(mH)}} \quad (6.4)$$

โดยที่ R_{fin} คือ ค่าความต้านทานความร้อนของครีบริบายความร้อน

h คือ ค่าสัมประสิทธิ์การพาความร้อนของอากาศ

$P = 2HL + 2Hd$ คือ พื้นที่ผิวรอบครีบริบายความร้อน

$A_c = dL$ คือ พื้นที่หน้าตัดของครีบริบายความร้อน

L คือ ความยาวของอุปกรณ์ระบายความร้อน (ดังแสดงในรูปที่ 6-8)

H คือ ความสูงของครีบริบายความร้อน (ดังแสดงในรูปที่ 6-8)

d คือ ความหนาของครีบริบายความร้อน (ดังแสดงในรูปที่ 6-7)

k คือ ค่าสัมประสิทธิ์การนำความร้อนของวัสดุที่ใช้ผลิตอุปกรณ์ระบายความร้อน

$$m = \sqrt{\frac{hP}{kA_c}}$$

สำหรับค่าความต้านทานความร้อนของฐานอุปกรณ์ระบายความร้อน มีค่าตามสมการที่ (6.5) ดังนี้

$$R_{base} = \frac{t_b}{kA_b} \quad (6.5)$$

โดยที่ R_{base} คือ ค่าความต้านทานความร้อนของฐานอุปกรณ์ระบายความร้อน

t_b คือ ความหนาของฐานอุปกรณ์ระบายความร้อน (ดังแสดงในรูปที่ 6-7)

$A_b = WL$ คือ พื้นที่ของฐานอุปกรณ์ระบายความร้อน

W คือ ความกว้างของอุปกรณ์ระบายความร้อน (ดังแสดงในรูปที่ 6-7)

ค่าความต้านทานความร้อนของพื้นที่ระหว่างครีบริบายความร้อน มีค่าตามสมการที่ (6.6) ดังนี้

$$R_b = hbL \quad (6.6)$$

โดยที่ R_b คือ ค่าความต้านทานความร้อนบริเวณช่องว่างระหว่างครีบริบายความร้อน

b คือ ระยะห่างระหว่างครีบริบายความร้อน (ดังแสดงในรูปที่ 6-7)

จากค่าความต้านทานความร้อนในสมการที่ (6.4) ถึงสมการที่ (6.6) เมื่อนำค่าที่ได้มารวมกันเพื่อหาค่าความต้านทานความร้อนรวมของอุปกรณ์ระบายความร้อน ได้ค่าความต้านทานความร้อนรวมตาม สมการที่ (6.7) ดังนี้

$$R_{HS} = \frac{1}{(N / R_{fin}) + h(N-1)bL} + \frac{t_b}{kLW} \quad (6.7)$$

โดยที่ N คือ จำนวนของครีประบายความร้อน

h คือ ค่าสัมประสิทธิ์การพาความร้อนของอากาศ

ค่าความต้านทานความร้อนรวมของอุปกรณ์ระบายความร้อนดังแสดงไว้ตามสมการที่ (6.7) มีตัวแปรไม่ทราบค่าคือสัมประสิทธิ์การพาความร้อนของอากาศที่ไหลเข้ามาระบายความร้อน ซึ่งในกรณีของการหาค่าของตัวแปรสัมประสิทธิ์การพาความร้อนนี้ ได้มีผู้ที่ได้ทำการศึกษาวิจัยเอาไว้ ซึ่งได้ค่าสมการสำหรับการหาค่าสัมประสิทธิ์การพาความร้อนของอากาศที่ไหลผ่านอุปกรณ์ระบายความร้อนแบบครีแบนดังนี้

$$Nu_b = \left[\left(\frac{Re_b^* Pr}{2} \right)^{-3} + \left(0.664 (Re_b^*)^{\frac{1}{2}} Pr^{\frac{1}{3}} \left(1 + \frac{3.65}{\sqrt{Re_b^*}} \right)^{-3} \right) \right]^{-\frac{1}{3}} \quad (6.8)$$

เมื่อ Nu_b คือ เลขนัสเซิลต์ (Nusselt Number) ซึ่งหาได้จากสมการที่ (6.9)

Pr คือ เลขพรันด์เทิล (Prandtl Number) ของอากาศ

Re_b^* คือ เลขเรย์โนลด์ของช่องทางการไหล (Channel Reynolds Number) หาได้จากสมการที่ (6.10)

$$Nu_b = \frac{hb}{k_f} \quad (6.9)$$

เมื่อ k_f คือ ค่าสัมประสิทธิ์การนำความร้อนของอากาศ

$$Re_b^* = Re_b \left(\frac{b}{L} \right) \quad (6.10)$$

เลขเรย์โนลด์จากสมการที่ (6.10) หาค่าได้ตามสมการที่ (6.11) ดังนี้

$$Re_b = \frac{b.V_{ch}}{\nu} \quad (6.11)$$

เมื่อ V_{ch} คือ ความเร็วของอากาศที่ไหลผ่านช่องว่างระหว่างครี หาได้จากสมการที่ (6.12)

ν คือ ความหนืดคินเนติกของอากาศ (Kinematic Viscosity)

$$V_{ch} = V_f \left(1 + \frac{d}{b}\right) \quad (6.12)$$

สมการที่ (6.4) ถึงสมการที่ (6.12) ใช้ในการหาค่าอุณหภูมิจุดต่อระหว่างอุปกรณ์ระบายความร้อนแบบครีปแผ่นกับอุปกรณ์อิเล็กทรอนิกส์ตามสมการที่ (6.3) ซึ่งถ้าอุปกรณ์ระบายความร้อนให้ค่าอุณหภูมิจุดต่อต่ำ แสดงว่าอุปกรณ์ระบายความร้อนนั้นมีสมรรถนะของการระบายความร้อนสูง แต่ในการออกแบบอุปกรณ์ระบายความร้อนด้วยอากาศนอกจากต้องคำนึงถึงสมรรถนะของการระบายความร้อนแล้ว ยังต้องคำนึงถึงการประหยัดพลังงานด้วย นั่นก็คือค่าพลังงานหรือกำลังที่ใช้ในการผลักอากาศเข้าไประบายความร้อนของพัดลมเป่าอากาศนั่นเอง ซึ่งค่ากำลังที่ใช้ในผลักอากาศของพัดลมระบายอากาศสามารถคำนวณได้ตามสมการต่อไปนี้

$$P_F = \frac{\dot{m}_a \Delta P}{\rho_a} = F_d V_f \quad (6.13)$$

666

โดยที่	P_F	คือ กำลังของพัดลมที่ใช้ในการผลักอากาศ
	ΔP	คือ ผลต่างความดันคร่อมอุปกรณ์ระบายความร้อน
	ρ_a	คือ ความหนาแน่นของอากาศ
	F_d	คือ แรงต้านการไหลของอากาศ
	V_f	คือ ความเร็วของอากาศที่เข้าไประบายความร้อน

แรงต้านการไหลของอากาศในสมการที่ (6.13) สามารถหาค่าได้จากการประยุกต์ใช้สมการที่ได้ ดังนี้

$$\frac{F_d}{\frac{1}{2} \rho_a V_{ch}^2} = f_{app} N(2HL + bL) + K_c(HW) + K_e(HW) \quad (6.14)$$

เมื่อ	f_{app}	คือ ค่าตัวประกอบความเสียดทานปรากฏ (Apparent Friction Factor)
	K_c	คือ สัมประสิทธิ์การหดตัวกระทันหัน (Sudden Contraction Coefficient)
	K_e	คือ สัมประสิทธิ์การขยายตัว (Expansion Coefficient)

สำหรับค่าตัวประกอบความเสียดทานปรากฏในการศึกษาครั้งนี้ได้ประยุกต์ใช้ผลการศึกษามีมาก่อนมาใช้อย่างสมการที่ (6.15) ส่วนสัมประสิทธิ์การหดตัวกระทันหันและสัมประสิทธิ์การขยายตัวแสดงไว้ในสมการที่ (6.17) และสมการ (6.18) ตามลำดับ

$$f_{app} \text{Re}_{D_h} = \left[\left(\frac{3.44}{\sqrt{L^*}} \right)^2 + (f \text{Re}_{D_h})^2 \right]^{\frac{1}{2}} \quad (6.15)$$

$$\text{เมื่อ } L^* = \frac{L}{D_h \text{Re}_{D_h}}, \text{Re}_{D_h} = \frac{V_{ch} D_h}{\nu}$$

$D_h = 2b$ คือ เส้นผ่านศูนย์กลางไฮดรอลิก (Hydraulic Diameter) ของช่องว่างระหว่างครีบริบายความร้อน

f คือ ตัวประกอบความเสียดทาน (Friction factor)

ค่าตัวประกอบความเสียดทานในสมการที่ (6.15) สามารถหาค่าได้จากการประยุกต์ใช้ผลการศึกษาที่มีมาก่อน ดังนี้

$$f \text{Re}_{D_h} = 24 - 32.527 \left(\frac{b}{H} \right) + 46.721 \left(\frac{b}{H} \right)^2 - 40.829 \left(\frac{b}{H} \right)^3 + 22.954 \left(\frac{b}{H} \right)^4 - 6.089 \left(\frac{b}{H} \right)^5 \quad (6.16)$$

$$K_c = 0.42(1 - \sigma^2) \quad (6.17)$$

$$K_e = (1 - \sigma^2)^2 \quad (6.18)$$

$$\text{โดยที่ } \sigma = 1 - \frac{Nd}{W}$$

จากการวิเคราะห์หาสมการการถ่ายเทความร้อนของอุปกรณ์ระบายความร้อนแบบครีบริบายแผ่นข้างต้นพบว่าสมการที่ได้มีความยุ่งยากและซับซ้อน หากจะนำสมการดังกล่าวไปใช้ในการออกแบบหาค่าเหมาะที่สุดจะทำให้เกิดความยุ่งยากในขั้นตอนกระบวนการออกแบบ แต่นอกจากการหาสมการระบายความร้อนของอุปกรณ์ระบายความร้อนโดยใช้สมการทางคณิตศาสตร์แล้วยังมีอีกหนึ่งวิธีที่นิยมใช้คือระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณซึ่งจะได้กล่าวในหัวข้อต่อไป

6.3 การวิเคราะห์การถ่ายเทความร้อนด้วยระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณ

พลศาสตร์ของไหลเชิงคำนวณ (Computational Fluid Dynamics; CFD) เป็นวิทยาการที่ใช้วิธีการวิเคราะห์เชิงตัวเลขและการเขียนโปรแกรมคอมพิวเตอร์ เพื่อศึกษาพฤติกรรมต่าง ๆ ของพลศาสตร์ของไหลและช่วยในการออกแบบทางอุตสาหกรรม การใช้วิธีการคำนวณโดยใช้โปรแกรมคอมพิวเตอร์ช่วยในการศึกษาและการออกแบบทางพลศาสตร์ของไหลนี้มีข้อได้เปรียบเมื่อเทียบกับการใช้วิธีทำการทดลองเนื่องจากสามารถศึกษาพฤติกรรมของของไหลในทุกสภาวะ และในทุกรูปทรงโดยใช้ค่าใช้จ่ายถูกกว่า และไม่มีความเสี่ยงต่ออันตราย ประกอบกับปัจจุบันคอมพิวเตอร์มีความสามารถในการคำนวณและแสดงผลที่สูงขึ้น เราจึงสามารถจำลองพฤติกรรมการไหลโดยใช้โปรแกรมคอมพิวเตอร์ทำหน้าที่ประมวลผลและแสดง

ค่าคุณสมบัติต่างๆ ของการไหล เช่น ความเร็ว ความดัน และอุณหภูมิออกมาให้เห็นได้ชัดเจนนอกจากนั้น การเปลี่ยนแปลงลักษณะชิ้นงานก็สามารถทำได้สะดวกรวดเร็วและใช้เวลาน้อยกว่า

ขั้นตอนการทำงานของพลศาสตร์ของไหลเชิงคำนวณโดยทั่ว ๆ ไป แบ่งได้เป็น 3 ส่วนคือ

1) ขั้นตอนก่อนการคำนวณ (Pre-Processor)

ขั้นตอนนี้เป็นการกำหนดลักษณะและเงื่อนไขต่าง ๆ ของปัญหาที่จะทำการวิเคราะห์ ขั้นตอนหลักก่อนการคำนวณมีดังนี้

- กำหนดลักษณะทางกายภาพของบริเวณที่สนใจ (Physical Domain)
- ทำการสร้างกริด (Grid) ภายในบริเวณที่สนใจโดยทำการแบ่งบริเวณที่สนใจออกเป็นส่วนย่อย ๆ
- กำหนดค่าสมบัติต่าง ๆ ของของไหลที่จะทำการคำนวณเช่น ค่าความหนาแน่น ความหนืด เป็นต้น
- กำหนดลักษณะของปัญหาเช่นเป็นการไหลแบบราบเรียบหรือปั่นป่วน
- กำหนดขอบเขตของปัญหา (Boundary Conditions)

2) ขั้นตอนการแก้ปัญหา (Solver)

เป็นขั้นตอนที่ใช้ระเบียบวิธีคำนวณเชิงตัวเลขเข้ามาหาคำตอบของปัญหาตามที่ได้กำหนดลักษณะและเงื่อนไขของปัญหาไว้ในขั้นตอนก่อนการคำนวณ สำหรับวิธีการแก้ปัญหาในการวิเคราะห์ด้วยระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณ มี 3 วิธีหลักคือ วิธีผลต่างอันตะ (Finite Difference Method: FDM), วิธีปริมาตรย่อยจำกัดหรือไฟไนท์วอลุ่ม (Finite Volume Method: FVM), และวิธีไฟไนต์เอลิเมนต์ (Finite Element Method: FEM) ซึ่งวิธีที่ถูกนำมาใช้มากที่สุดก็คือวิธีปริมาตรจำกัด โดยในการหาคำตอบของปัญหาจะต้องแก้สมการที่เกี่ยวข้องเพื่อหาคำตอบคือสมการอนุรักษ์มวล (Conservation of Mass) ซึ่งจากการวิเคราะห์จะได้สมการความต่อเนื่อง (Continuity Equation) ดังแสดงไว้ในสมการที่ (6.19), สมการอนุรักษ์พลังงาน (Conservation of Energy) ดังแสดงไว้ในสมการที่ (6.20) สมการของโมเมนตัม (Momentum Equations) ซึ่งได้จากการวิเคราะห์กฎข้อที่ 2 ของนิวตัน (Newton's Second Law) ดังแสดงไว้ในสมการที่ (6.22) และสมการพลังงาน (6.23)

$$\frac{1}{\rho} \left(\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} \right) + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0 \quad (6.19)$$

$$\rho \frac{Dh}{Dt} - \frac{Dp}{Dt} = \Phi + \frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) \quad (6.20)$$

โดยที่

$$\begin{aligned} \Phi = & 2\mu \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 + \frac{1}{2} \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right)^2 \right] \\ & - \frac{2}{3} \mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right)^2 \end{aligned} \quad (6.21)$$

$$\begin{aligned}
\rho \frac{Du}{Dt} &= \rho f_x - \frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \\
\rho \frac{Dv}{Dt} &= \rho f_y - \frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} \\
\rho \frac{Dw}{Dt} &= \rho f_z - \frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z}
\end{aligned} \tag{6.22}$$

$$\frac{\partial(\rho h_0)}{\partial t} + \nabla \cdot (\rho h_0 \vec{V}) = -p \nabla \cdot \vec{V} + \nabla \cdot (k \nabla T) + \Phi + S_h \tag{6.23}$$

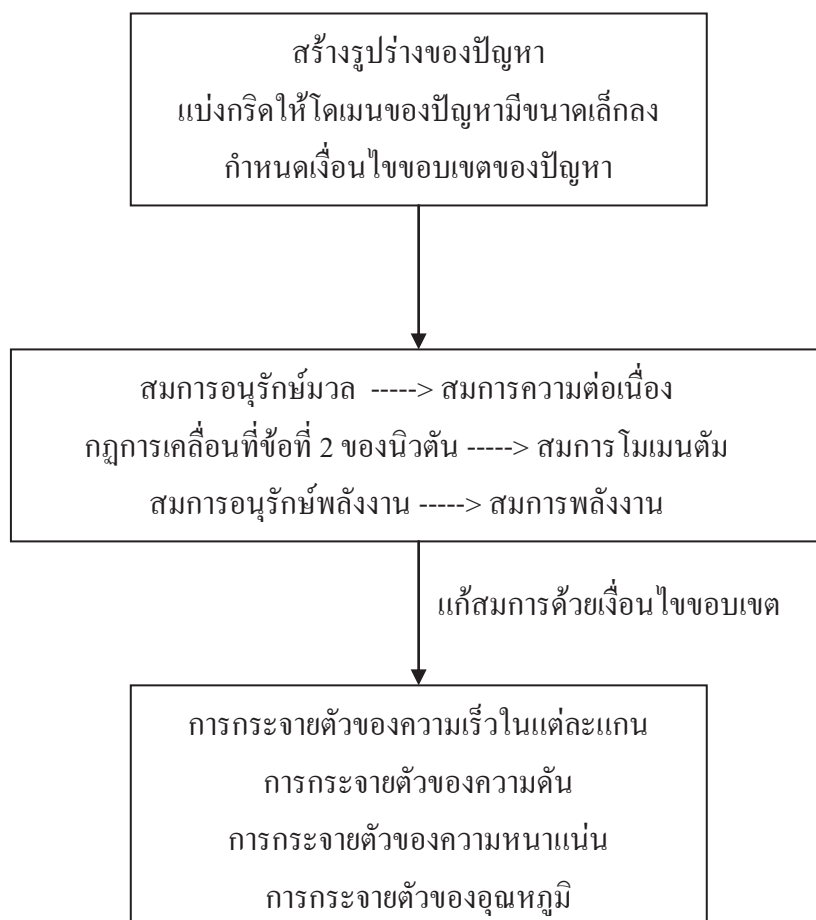
โดยคำตอบที่ได้จากการแก้สมการข้างบนคือความเร็วในแนวแกน x (v), ความเร็วในแนวแกน y (u), ความเร็วในแนวแกน z (w), อุณหภูมิ (T), ความดัน (p) และความหนาแน่นของของไหล (ρ) นั่นเอง

3) ขั้นตอนหลังการแก้ปัญหา (Post-Processor)

เป็นขั้นตอนในการแสดงผลที่ได้จากขั้นตอนการแก้ปัญหาโดยอาจจะแสดงออกในรูปของตัวเลข รูปภาพหรือกราฟที่แสดงค่าต่าง ๆ หรือเวกเตอร์ของตัวแปรที่ผู้ศึกษาสนใจ

ขั้นตอนการทำงานของระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณแสดงไว้ดังรูปที่ 6-9

ในการศึกษาครั้งนี้เลือกใช้ระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณในการหาคำตอบของฟังก์ชันเป้าหมายของการออกแบบหาค่าที่เหมาะสมที่สุด และเลือกใช้วิธีปริมาตรจำกัดในการหาคำตอบของการวิเคราะห์หาค่าตัวแปรที่เกี่ยวข้องต่าง ๆ ของปัญหา เนื่องจากรูปสมการที่ได้จากการใช้วิธีปริมาตรจำกัดจะแสดงให้เห็นถึงความสัมพันธ์ของคุณสมบัติต่างๆ ในแง่ของการอนุรักษ์ภายในแต่ละปริมาตรจำกัด ซึ่งความสัมพันธ์ที่เห็นได้อย่างเด่นชัดระหว่างวิธีการทางตัวเลขกับหลักการอนุรักษ์เชิงฟิสิกส์เป็นสิ่งที่ทำให้วิธีปริมาตรจำกัดเป็นที่นิยมใช้สำหรับการแก้ปัญหาทางพลศาสตร์ของไหล เพราะถูกต้องตามหลักฟิสิกส์และง่ายที่จะทำความเข้าใจมากกว่าวิธีผลต่างอันดับหรือวิธีไฟไนต์เอลิเมนต์



รูปที่ 6-9 แสดงขั้นตอนการทำงานของระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณ

6.4 สมมติฐานของการศึกษา

เนื่องจากในการวิเคราะห์หาค่าฟังก์ชันเป้าหมายในการออกแบบหาค่าที่เหมาะสมที่สุดของอุปกรณ์ระบายความร้อนด้วยอากาศแบบครีปแผ่นที่ถูกนำมาใช้เป็นกรณีศึกษานี้ ใช้ระเบียบวิธีพลศาสตร์ของไหลเชิงคำนวณ เพื่อความสะดวกในการวิเคราะห์จึงมีการตั้งสมมติฐานไว้ดังนี้

1. การไหลของอากาศเป็นแบบราบเรียบและคงตัว
2. สมบัติเชิงความร้อนของวัสดุทั้งอากาศและของแข็งคงที่ไม่ขึ้นกับอุณหภูมิ
3. ความเร็วของอากาศเข้าอุปกรณ์ระบายความร้อนกระจายตัวทั้งพื้นที่อย่างสม่ำเสมอ
4. ความร้อนกระจายตัวอย่างสม่ำเสมอตลอดพื้นผิวฐานของอุปกรณ์ระบายความร้อน

ตัวอุปกรณ์ระบายความร้อนทำจากอะลูมิเนียม สมบัติทางกายภาพของของแข็งและของไหลที่ความดัน 1 บรรยากาศและอุณหภูมิ 25°C แสดงไว้ในตารางที่ 6-1

ตารางที่ 6-1 สมบัติของอลูมิเนียมและอากาศ

สมบัติของอลูมิเนียม	สมบัติของอากาศ
ความหนาแน่น = 2719 kg/m^3 ความร้อนจำเพาะ = 871 J/kg-K สัมประสิทธิ์การนำความร้อน = 202 W/m-K	ความหนาแน่น = 1.1770 kg/m^3 ความร้อนจำเพาะ = 1006 J/kg-K สัมประสิทธิ์การนำความร้อน = 0.0267 W/m-K ความหนืด = $1.8832 \times 10^{-5} \text{ kg/m-s}$

6.5 ฟังก์ชันเป้าหมายและเงื่อนไขของการหาค่าเหมาะที่สุด

ฟังก์ชันเป้าหมายที่ใช้ในการออกแบบหาค่าเหมาะที่สุดของอุปกรณ์ระบายความร้อน ในการศึกษาครั้งนี้มีทั้งหมดสองฟังก์ชันเป้าหมายคือ ค่าอุณหภูมิจุดต่อระหว่างอุปกรณ์ระบายความร้อนและอุปกรณ์อิเล็กทรอนิกส์ต่ำสุด และค่ากำลังที่พัดลมต้องใช้ในการผลักอากาศเข้าไประบายความร้อนต่ำสุด ซึ่งทั้งสองเป้าหมายนี้มีความขัดแย้งกันกล่าวคือหากต้องการให้ค่าอุณหภูมิจุดต่อต่ำจะต้องใช้ความเร็วของอากาศในการระบายความร้อนสูงซึ่งจะส่งผลให้ต้องใช้กำลังในการผลักอากาศสูงตามไปด้วย ในทางกลับกันเมื่อใช้ความเร็วของอากาศต่ำเพื่อเป็นการลดกำลังของพัดลมลง ผลที่ตามมาคือการระบายความร้อนออกจากอุปกรณ์อิเล็กทรอนิกส์ทำได้น้อยส่งผลให้อุณหภูมิจุดต่อมีค่าสูงขึ้น สำหรับสมการการหาค่าเหมาะที่สุดแบบหลายเป้าหมายเป็นไปตามสมการที่ (6.24)

$$\min_x f(x) = \{f_1(x) \quad f_2(x)\} \quad (6.24)$$

เมื่อฟังก์ชันเป้าหมาย f_1 และ f_2 มีค่าตามสมการที่ (6.25) และสมการที่ (6.26) ดังนี้

$$f_1(x) = T_{jc} = T_a + QR_{HS} \quad (6.25)$$

$$f_2(x) = P_F = \frac{\dot{m}_a \Delta P}{\rho_a} \quad (6.26)$$

ในการออกแบบหาค่าเหมาะที่สุดได้กำหนดเงื่อนไขของการออกแบบ ซึ่งเป็นเงื่อนไขของรูปร่างอุปกรณ์ระบายความร้อนและเงื่อนไขอุณหภูมิที่จุดต่อไว้ดังต่อไปนี้

$$10 - \frac{H_{\min}}{b} \leq 0$$

$$\frac{H_{\max}}{b} - 25 \leq 0$$

$$f_1 - 355 \leq 0$$

โดยที่	$H_{\min}$	คือ ความสูงต่ำสุดของครีประบายความร้อน
	$H_{\max}$	คือ ความสูงสูงสุดของครีประบายความร้อน
	b	คือ ระยะห่างระหว่างครีประบายความร้อน

การออกแบบอุปกรณ์ระบายความร้อนแบบครีปแผ่นตรง โดยใช้ SPEA ร่วมกับแบบจำลองพื้นผิวดตอบสนอง

7.1 การทำงานร่วมกันของSPEAและแบบจำลองพื้นผิวดตอบสนอง

การทำงานร่วมกันของขั้นตอนวิธีวิวัฒนาการSPEAและแบบจำลองพื้นผิวดตอบสนองในการศึกษานี้ มีรูปแบบคล้ายกับรูปแบบการผสมผสานแบบที่ 3.3.3 และนำเสนอเป็นขั้นตอนได้ดังนี้

- 1: เริ่มต้นการทำงาน กำหนดให้ค่า $k=0$, กำหนดประชากรต้นแบบให้มีค่าเท่ากับ $P^k = \{x_1, x_2, \dots, x_N\}$, หาค่าฟังก์ชันเป้าหมายของการออกแบบ $f(P^k)$, และกำหนดค่าสมาชิกของเมตริกซ์ผลเฉลยภายนอกของพาริโต A^k
- 2: เลือกสมาชิกบางตัวของประชากรต้นแบบ P^k และค่าฟังก์ชันเป้าหมายของสมาชิกนั้น ๆ $f_i(k)$ เพื่อนำไปสร้างแบบจำลองพื้นผิวดตอบสนอง
- 3: หาค่าแนวของพาริโตโดยใช้ฟังก์ชันเป้าหมายจากแบบจำลองการประมาณค่าของวิธีพื้นผิวดตอบสนอง โดยมีขั้นตอนย่อยดังนี้

3.1 เริ่มต้นการทำงาน: กำหนดให้ $t=0$, ประชากรต้นแบบ $Q^t = [y_1, y_2, \dots, y_N]$, หาค่าฟังก์ชันเป้าหมาย $g(Q^t)$, และกำหนดค่าสมาชิกของเมตริกซ์เก็บผลเฉลยภายนอกของพาริโต $B^t = \{ \}$

3.2 กำหนดค่าความเหมาะสมของฟังก์ชัน (Fitness Values) G^t ให้กับสมาชิกของผลรวมของเซตประชากรต้นแบบและเมตริกซ์ผลเฉลยภายนอกของพาริโต ($Q^t \cup B^t$) เพื่อทำการคัดเลือกประชากร

3.3 ปรับปรุงเมตริกซ์เก็บผลเฉลยของพาริโตด้วยวิธีการคัดเลือกประชากรของ ($Q^t \cup B^t$) ด้วยวิธีตามสิ่งแวดล้อม โดยใช้ค่า G^t จากขั้นตอนที่ 3.2 ผลที่ได้จากการคัดเลือกคือเมตริกซ์เก็บผลเฉลยของพาริโตใหม่ B^{t+1}

3.4 ถ้าจำนวนสมาชิกของ B^{t+1} มากกว่าค่าที่กำหนดไว้ให้รับได้ ใช้วิธี nearest neighbourhood (NNT) ในการกำจัดสมาชิกบางตัวทิ้งไป

3.5 คัดเลือกประชากรต้นแบบโดยใช้การเลือกด้วยการแข่งขันแบบเลขฐานสอง (Binary Tournament Selection) จาก B^{t+1} ผลที่ได้คือเมตริกซ์ประชากรของตัวแปรออกแบบของรอบการคำนวณที่ t^{th} , $Q_1^t = select(B^{t+1})$

3.6 แลกเปลี่ยนยีนส์ของประชากรของตัวแปรออกแบบใน Q_1^t ผลที่ได้คือเมตริกซ์ประชากรของตัวแปรออกแบบของรอบการคำนวณที่ t^{th} , $Q_2^t = recombine(Q_1^t)$

3.7 มิวเตชัน: $Q_2^{t+1} = mutate(Q_2^t)$

3.8 หาค่าฟังก์ชันเป้าหมายที่ได้จากแบบจำลองพื้นผิวดอปสันอง, $g(Q^{t+1})$, โดยใช้ประชากรตัวแปรออกแบบที่ได้จากขั้นตอนที่ 3.7

3.9 หยุดการทำงานแล้วไปที่ขั้นตอนที่ 3.10 ถ้าวนรอบการทำงานครบตามที่กำหนดไว้ $t = t_{\max}$, หรือถ้ายังวนรอบไม่ครบตามที่กำหนดไว้ กำหนดให้ $t = t + 1$ แล้วไปที่ขั้นตอนที่ 3.2

3.10 บันทึกค่าผลเฉลยที่ไม่ถูกรอรับ (จากการเปรียบเทียบค่าของขั้นตอนที่ 3.8) ไว้ในเมตริกซ์เก็บผลเฉลยภายนอกของพารโต B^{t+1} แล้วส่งค่าไปที่ขั้นตอนที่ 4

4: หาค่าฟังก์ชันเป้าหมายของการออกแบบโดยใช้ฟังก์ชันการออกแบบจริงจากประชากรของตัวแปรออกแบบใน B^{t+1} , $f(B^{t+1})$

5: กำหนดค่าความเหมาะสมของฟังก์ชัน (Fitness Values) F^k ให้กับสมาชิกของผลรวมของเซตประชากรต้นแบบเมตริกซ์ผลเฉลยภายนอกของพารโต และเมตริกซ์ผลเฉลยภายนอกของพารโตที่ได้จากการหาค่าเหมาะสมของแบบจำลองพื้นผิวดอปสันอง ($P^k \cup A^k \cup B^{t+1}$) เพื่อทำการคัดเลือกประชากร

6: ปรับปรุงเมตริกซ์เก็บผลเฉลยของพารโตด้วยวิธีการคัดเลือกประชากรของ ($P^k \cup A^k \cup B^{t+1}$) ด้วยวิธีตามสิ่งแวดล้อม โดยใช้ค่า F^k จากขั้นตอนที่ 5 ผลที่ได้จากการคัดเลือกคือเมตริกซ์เก็บผลเฉลยของพารโตใหม่ A^{k+1}

7: ถ้าจำนวนสมาชิกของ A^{k+1} มากกว่าค่าที่กำหนดไว้ให้รับได้ ให้ใช้วิธีกรองผลเฉลยกำจัดสมาชิกบางตัวทิ้งไป

8: คัดเลือกประชากรต้นแบบโดยใช้การเลือกด้วยการแข่งขันแบบเลขฐานสองจาก A^{k+1} ผลที่ได้คือเมตริกซ์ประชากรของตัวแปรออกแบบของรอบการคำนวณที่ k^{th} , $P_1^k = select(A^{k+1})$

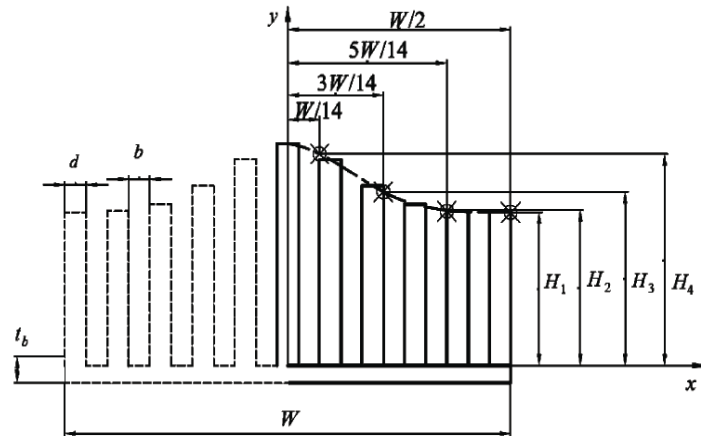
9: แลกเปลี่ยนยีนส์ของประชากรของตัวแปรออกแบบใน P_1^k ผลที่ได้คือเมตริกซ์ประชากรของตัวแปรออกแบบของรอบการคำนวณที่ k^{th} , $P_2^k = recombine(P_1^k)$

10: มิวเตชัน: $P^{k+1} = mutate(P_2^k)$

11: หาค่าฟังก์ชันเป้าหมาย, $f(P^{k+1})$, โดยใช้ประชากรตัวแปรออกแบบที่ได้จากขั้นตอนที่ 10

12: หยุดการทำงาน ถ้าวนรอบการทำงานครบตามที่กำหนดไว้ $k = k_{\max}$, หรือถ้ายังวนรอบไม่ครบตามที่กำหนดไว้ กำหนดให้ $k = k + 1$ แล้วไปที่ขั้นตอนที่ 5

ในการศึกษาเพื่อออกแบบหาค่าเหมาะสมที่สุดของตัวแปรรูปร่างของอุปกรณ์ระบายความร้อนแบบครีบนแผ่นของบทที่ 7 แตกต่างจากงานวิจัยที่ทำมาก่อนคือ การเพิ่มตัวแปรกำหนดการกระจายความสูงของครีบน (ดังแสดงไว้ในรูปที่ 7-1) เพื่อให้การกระจายความสูงของครีบนระบายความร้อนมีมากขึ้น สำหรับค่าความกว้างและความยาวของอุปกรณ์ระบายความร้อนกำหนดให้มีค่าเท่ากับ 60 และ 80 มิลลิเมตรตามลำดับ



รูปที่ 7-1 ตัวแปรออกแบบของปัญหาและลักษณะการกระจายความสูงของครีบ

สำหรับฟังก์ชันขั้นตอนการทำงานร่วมกันของ SPEA และแบบจำลองพื้นผิวดตอบสนองนำเสนอไว้ในรูปที่ 7-2 และเพื่อให้การกำหนดค่าตัวแปรออกแบบและการทำงานร่วมกันของโปรแกรม MATLAB และโปรแกรมพลศาสตร์ของไหลเชิงคำนวณซึ่งใช้เป็นเครื่องมือในการหาค่าฟังก์ชันเป้าหมายของการออกแบบหาค่าที่เหมาะสมที่สุดในงานนี้ จึงได้มีการกำหนดการปรับค่าตัวแปรบางตัวเพื่อให้เกิดความเหมาะสมก่อนที่จะส่งค่าตัวแปรไปดำเนินการออกแบบในขั้นตอนต่อไป ซึ่งรายละเอียดการปรับค่าตัวแปรและการหาค่าฟังก์ชันเป้าหมายเป็นดังนี้

- 1: ป้อนค่าเวกเตอร์ของตัวแปรออกแบบ $X = [x_1, \dots, x_8]$
- 2: กำหนดค่า $d = x_1$, $n = \text{round}(x_2)$, $t = x_3$, $H_1 = x_4$, $H_2 = x_5$, $H_3 = x_6$, $H_4 = x_7$, $V_f = x_8$
- 3: หาค่าระยะห่างระหว่างครีบระบายความร้อนโดยใช้สมการดังนี้

$$b = \frac{W - nd}{n - 1}$$

- 4: ถ้าระยะห่างระหว่างครีบ b แคบหรือกว้างมาก ($b \notin [0.5, 5] \text{ mm}$) ให้เพิ่มหรือลดค่าจำนวนครีบจนกว่าระยะห่างระหว่างครีบ b จะเข้ามาอยู่ในช่วงที่กำหนดไว้

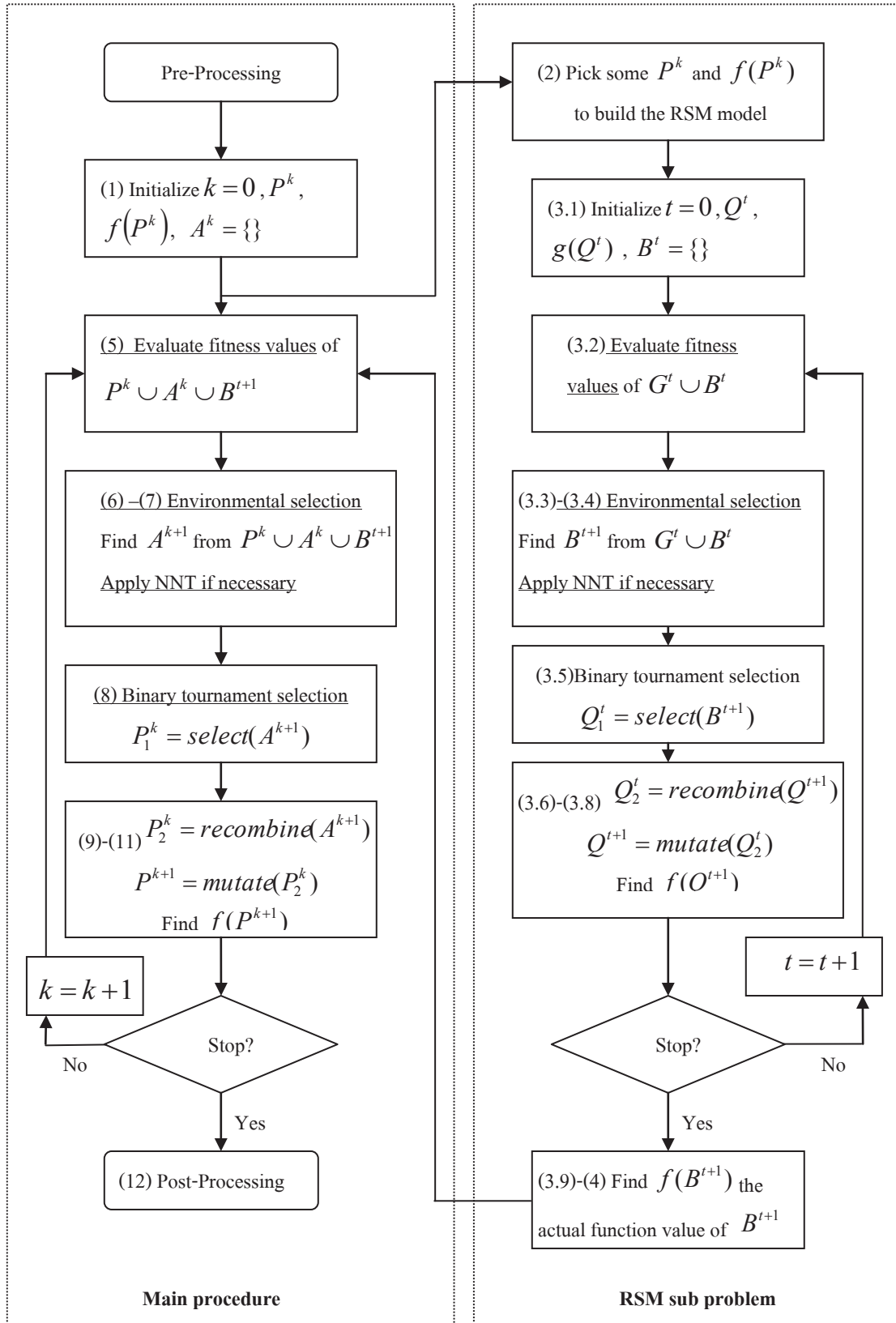
- 5: ประมวลค่ากลางของตัวแปรอิสระ $\left\{ \frac{W}{14}, \frac{3W}{14}, \frac{5W}{14}, \frac{W}{2} \right\}$ และตัวแปรตาม $\{H_1, H_2, H_3, H_4\}$

โดยใช้การประมวลค่ากลางแบบ Cubic Spline

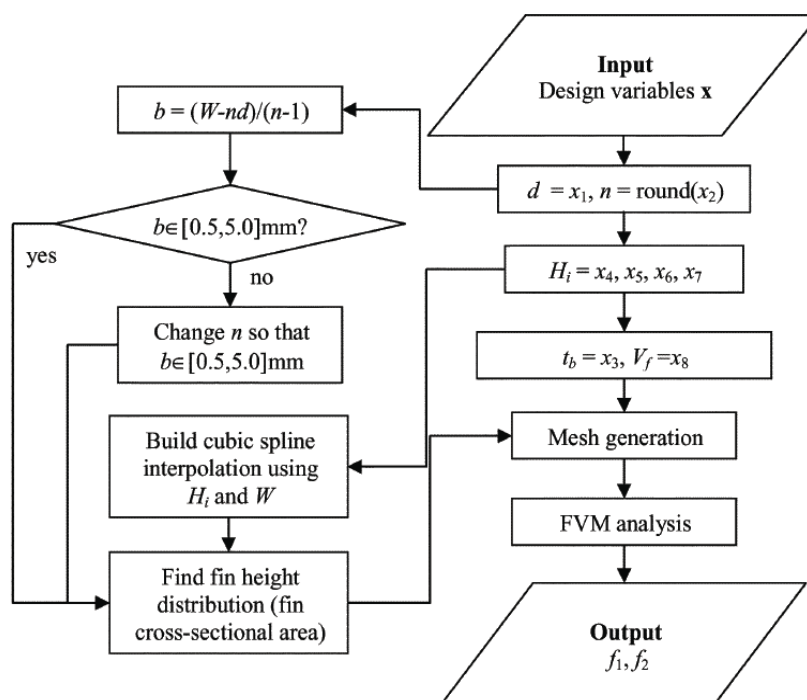
- 6: สร้างพื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนโดยใช้ตัวแปรทั้งหมดที่กล่าวมาข้างต้น
- 7: สร้างอินพุตไฟล์สำหรับป้อนให้กับโปรแกรมพลศาสตร์ของไหลเชิงคำนวณ
- 8: วิเคราะห์หาค่าต่าง ๆ ด้วยโปรแกรมพลศาสตร์ของไหลเชิงคำนวณ
- 9: อ่านค่าเอาต์พุตไฟล์ที่ได้จากข้อ 8 และนำไปคำนวณหาค่าฟังก์ชันเป้าหมาย f_1 และ f_2 เพื่อนำไปป้อนให้กับกระบวนการหาค่าที่เหมาะสมที่สุดต่อไป

สำหรับผังงานการดำเนินการปรับค่าตัวแปรออกแบบและการหาค่าฟังก์ชันเป้าหมายแสดงไว้ในรูปที่

7-3



รูปที่ 7-2 ผังการทำงานร่วมกันของSPEAและแบบจำลองพื้นผิวดตอบสนอง



รูปที่ 7-3 แสดงขั้นตอนการหาค่าฟังก์ชันเป้าหมายของการออกแบบ

ในการออกแบบหาค่าเหมาะที่สุดแบบหลายเป้าหมายด้วย SPEA โดยไม่ใช่แบบจำลองพื้นผิวดตอบสนอง (ในที่นี้ขอเรียกว่า “SPEA”) กำหนดให้มีขนาดของประชากรเท่ากับ 25 และใช้จำนวนรอบในการหาแนวผลเฉลยของพารามิเตอร์ทั้งหมด 50 รอบ สำหรับการออกแบบหาค่าเหมาะที่สุดด้วย SPEA ร่วมกับแบบจำลองพื้นผิวดตอบสนอง (ในที่นี้ขอเรียกว่า “SPEA-RSM”) กำหนดให้มีขนาดของประชากรเท่ากับ 25 และขนาดของเมตริกซ์เก็บผลเฉลยภายนอกของพารามิเตอร์ B' มีขนาดเท่ากับ 25 นั้นหมายความว่าในแต่ละรอบของการหาผลเฉลย กำหนดให้ 25 ผลเฉลยที่ไม่ถูกรอบงำที่ได้จากการหาผลเฉลยโดยใช้ฟังก์ชันเป้าหมายที่ได้จากการประมาณค่าด้วยแบบจำลองพื้นผิวดตอบสนองส่งเข้าไปรวมกับ 25 ผลเฉลยที่ไม่ถูกรอบงำของการหาผลเฉลยโดยฟังก์ชันเป้าหมายจริงด้วย SPEA เพื่อนำไปหาผลเฉลยของปัญหาการออกแบบต่อไป ดังนั้นจึงกำหนดค่าการวนรอบของ SPEA-RSM ไว้ที่ 25 รอบการคำนวณ (รวมแล้วจะได้ 50 รอบ) ซึ่งจะทำให้ได้จำนวนรอบของการวิเคราะห์ด้วยวิธีปริมาตรจำกัดเท่ากับจำนวนรอบที่ต้องวิเคราะห์ของ SPEA ขนาดของเมตริกซ์เก็บผลเฉลยภายนอกของพารามิเตอร์ A^k กำหนดไว้ที่ 80 ค่าทั้งสองวิธีการออกแบบ โดยในการเริ่มต้นทำงานของทั้งสองวิธีเริ่มต้นประชากรต้นแบบเดียวกัน กำหนดให้ค่าความเป็นไปได้ของการข้ามสายพันธุ์ (Crossover) และความเป็นไปได้ของการเกิดการมิวเตชันไว้ที่ 1.0 และ 0.5 ตามลำดับ

กำหนดขนาดจำนวนประชากรและจำนวนรอบของการวนรอบหาค่าตอบไว้ที่ 25 และ 50 ตามลำดับ ดังนั้นจำนวนครั้งที่จะต้องวิเคราะห์ด้วยวิธีปริมาตรจำกัดจะมีค่าเป็น 25×50 ครั้งต่อวิธีการออกแบบ ซึ่งจาก

การศึกษาจะเห็นได้ชัดเจนว่า SPEA-RSM จะใช้เวลาในการคำนวณมากกว่าเนื่องจากการคำนวณย่อยของแบบจำลองพื้นผิวดตอบสนองด้วย

7.2 ผลการศึกษา

เพื่อการอธิบายผลที่ได้จากการศึกษา จึงขอกำหนดให้กลุ่มของผลเฉลยที่ไม่ถูกรอรับที่ได้จากระบวนการหาค่าเหมาะที่สุดเป็นตัวแปรต่างๆ ดังต่อไปนี้

SPEA05 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA หลังจากการคำนวณรอบที่ 5

SPEA10 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA จากการคำนวณรอบที่ 10

SPEA20 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA จากการคำนวณรอบที่ 20

SPEA30 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA จากการคำนวณรอบที่ 30

SPEA40 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA จากการคำนวณรอบที่ 40

SPEA50 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA จากการคำนวณรอบที่ 50

SPEA-RSM01 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA-RSM จากการคำนวณรอบที่ 1

SPEA-RSM05 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA- RSM จากการคำนวณรอบที่ 5

SPEA-RSM10 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA- RSM จากการคำนวณรอบที่ 10

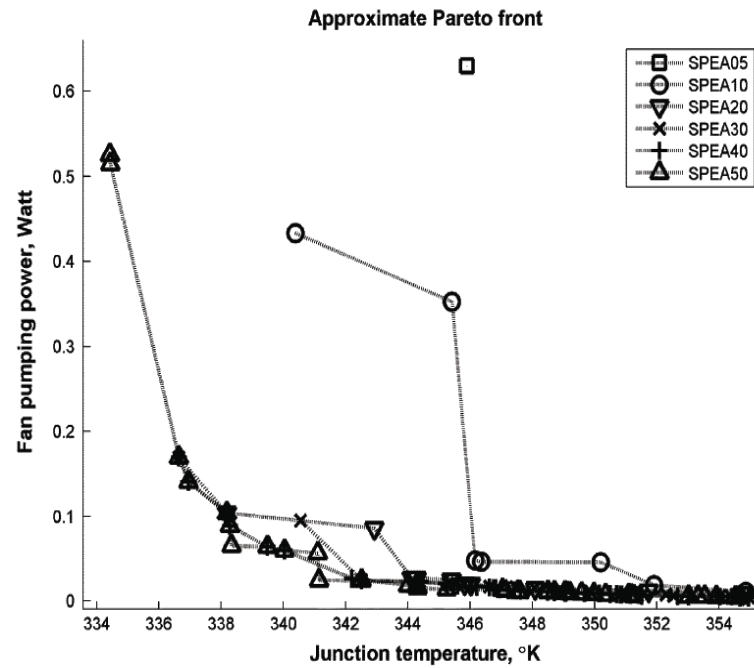
SPEA-RSM15 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA- RSM จากการคำนวณรอบที่ 15

SPEA-RSM20 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA- RSM จากการคำนวณรอบที่ 20

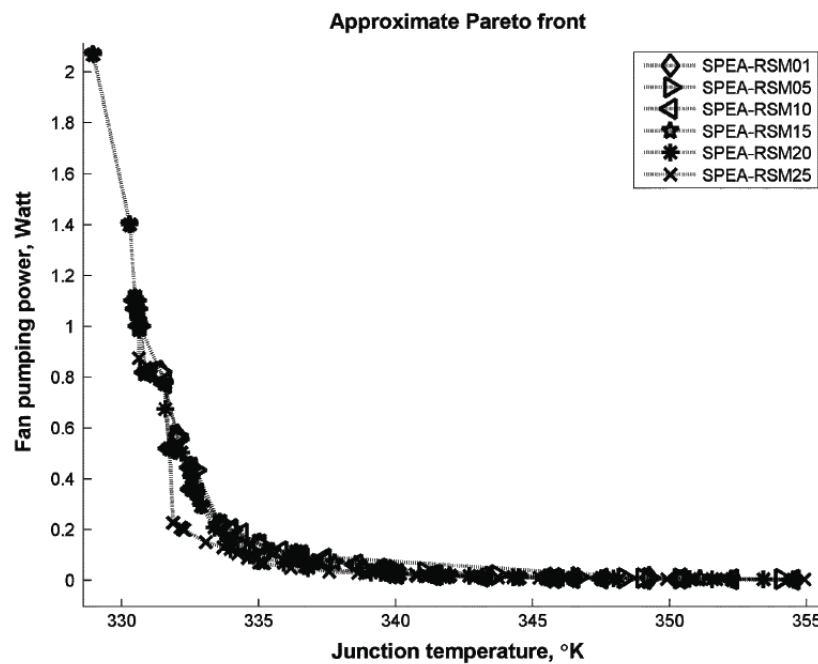
SPEA-RSM25 คือ แนวผลเฉลยที่ไม่ถูกรอรับของ SPEA- RSM จากการคำนวณรอบที่ 25

รูปที่ 7-4 แสดงวิวัฒนาการของแนวของผลเฉลยที่ไม่ถูกรอรับที่ได้จากการออกแบบโดยใช้ SPEA โดยในการศึกษาได้ตัดผลเฉลยที่ให้ค่าฟังก์ชันเป้าหมายที่หนึ่ง (f_1) หรือค่าอุณหภูมิจุดต่อของอุปกรณ์ระบายความร้อนกับหน่วยประมวลผลกลางมากกว่า $355^{\circ}K$ ทิ้งไป ซึ่งจากรูปที่ 7-4 พบว่าแนวของผลเฉลยที่ไม่ถูกรอรับจะมีการเปลี่ยนแปลงอย่างรวดเร็วในช่วงของการคำนวณเริ่มต้นและการเปลี่ยนแปลงเล็กน้อยที่รอบการคำนวณสุดท้ายคือรอบการคำนวณที่ 50

วิวัฒนาการของการหาผลเฉลยที่ไม่ถูกรอรับจากการออกแบบหาค่าเหมาะที่สุดโดยใช้ SPEA-RSM แสดงไว้ในรูปที่ 7-5 ซึ่งพบว่าแนวของผลเฉลยที่ไม่ถูกรอรับที่ได้จากรอบการคำนวณต่างๆ กันจะมีแนวโน้มใกล้เคียงกันหรือแตกต่างกันเพียงเล็กน้อยเท่านั้น



รูปที่ 7-4 แสดงแนวของพารेटอของ SPEA



รูปที่ 7-5 แสดงแนวของพารेटอของ SPEA-RSM

สมรรถนะของการออกแบบโดยใช้ SPEA และ SPEA-RSM ถูกนำมาเปรียบเทียบกันเพื่อหาวิธีการออกแบบที่ดีที่สุด โดยตัวแปรที่นำมาใช้ในการเปรียบเทียบของการศึกษาค้นคว้าครั้งนี้ คือตัวบ่งชี้ C (C -indicator) [90] ซึ่งความหมายของตัวบ่งชี้ C ที่นำมาเปรียบเทียบอธิบายได้ดังนี้

กำหนดให้ $C(A, B)$ คือตัวบ่งชี้ C สัมพัทธ์ของแนวผลเฉลยที่ไม่ถูกรอบงำของ B เทียบกับแนวผลเฉลยที่ไม่ถูกรอบงำของ A โดยหาได้จากการนำเอาจำนวนผลเฉลยในแนวผลเฉลยของ B ที่ถูกรอบงำหรือเท่ากับผลเฉลยของ A หารด้วยจำนวนของผลเฉลยของ B ทั้งหมด เขียนแทนได้ด้วยสมการดังนี้

$$C(A, B) = \frac{|\{b \in B; \exists a \in A : a \prec b\}|}{|B|} \quad (7.1)$$

เมื่อ A, B คือ แนวผลเฉลยของพาราโบลใด ๆ

a และ b คือ สมาชิกของแนวผลเฉลยของพาราโบล A และ B ตามลำดับ

$|B|$ คือ ขนาดของสมาชิกในแนวผลเฉลยของพาราโบล B

กำหนดให้ $C(B, A)$ คือ ตัวบ่งชี้ C สัมพัทธ์ของแนวผลเฉลยที่ไม่ถูกรอบงำของ A เทียบกับแนวผลเฉลยที่ไม่ถูกรอบงำของ B โดยหาได้จากการนำเอาจำนวนผลเฉลยในแนวผลเฉลยของ A ที่ถูกรอบงำหรือเท่ากับผลเฉลยของ B หารด้วยจำนวนผลเฉลยของ A ทั้งหมด เขียนแทนได้ด้วยสมการดังนี้

$$C(B, A) = \frac{|\{a \in A; \exists b \in B : b \prec a\}|}{|A|} \quad (7.2)$$

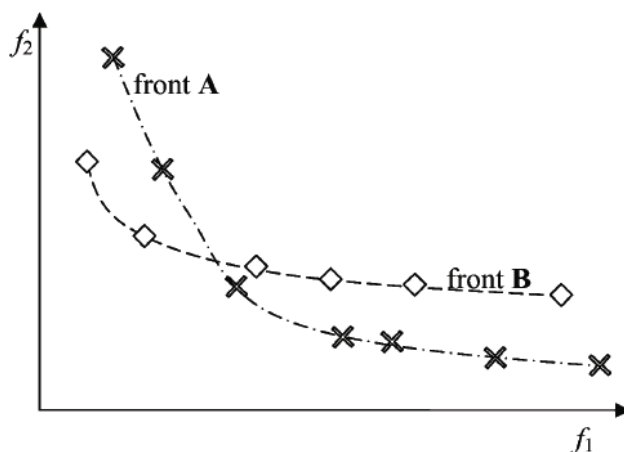
เมื่อ $|A|$ คือ ขนาดของสมาชิกในแนวผลเฉลยของพาราโบล A

ซึ่งจากค่าตัวบ่งชี้ C ในสมการที่ (7.1) และสมการที่ (7.2) สามารถแปลความหมายของการเปรียบเทียบได้ดังนี้

ถ้า $C(A, B) = 1$ หมายความว่า ผลเฉลยทั้งหมดในแนวผลเฉลย B ถูกรอบงำหรือเท่ากับผลเฉลย A

ถ้า $C(A, B) = 0$ หมายความว่า ไม่มีสมาชิกตัวใดในผลเฉลย B ถูกรอบงำหรือเท่ากับผลเฉลย A

รูปที่ 7-6 แสดงการเปรียบเทียบของแนวผลเฉลยที่ไม่ถูกรอบงำสองแนว คือ แนวผลเฉลยที่ไม่ถูกรอบงำ A และแนวผลเฉลยที่ไม่ถูกรอบงำ B โดยที่ผลเฉลยของ A และ B มีค่าเท่ากับ 7 และ 6 ตามลำดับ ตัวอย่างการหาค่าตัวบ่งชี้ C เปรียบเทียบกันของทั้งสองแนวผลเฉลย $C(A, B) = \frac{4}{6}$ จากกราฟรูปที่ 7-6 จะพบว่าจำนวนของผลเฉลยทั้งหมดของ B มีอยู่ด้วยกัน 6 ตัว และตัวที่ถูกรอบงำหรือเท่ากับผลเฉลย A มีทั้งหมด 4 ตัว ดังนั้น $C(A, B)$ จึงมีค่าเท่ากับ $\frac{4}{6}$, $C(B, A) = \frac{2}{7}$ จากกราฟรูปที่ 7-6 จะพบว่าจำนวนของผลเฉลยทั้งหมดของ A มีอยู่ด้วยกัน 7 ตัวและตัวที่ถูกรอบงำหรือเท่ากับผลเฉลย B มีทั้งหมด 2 ตัว ดังนั้น $C(B, A)$ จึงมีค่าเท่ากับ $\frac{2}{7}$ ซึ่งหมายความว่าแนวของผลเฉลย A ดีกว่าแนวของผลเฉลย B

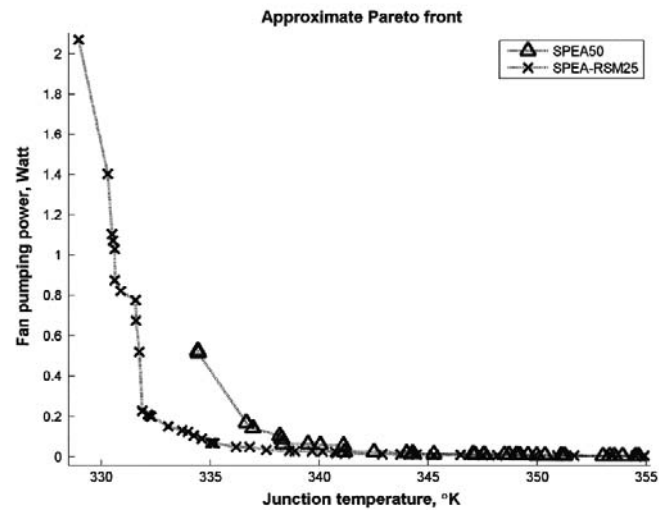


รูปที่ 7-6 แสดงตัวอย่างแนวของผลเฉลย A และ B เพื่อหาค่าตัวบ่งชี้ C

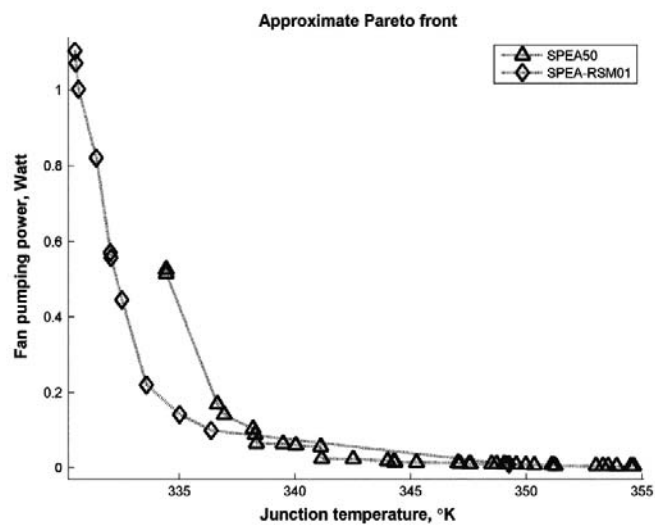
การเปรียบเทียบแนวของผลเฉลยที่ถูกครอบงำระหว่างการใช้ SPEA และ SPEA-RSM ซึ่งทั้งสองวิธีการหาค่าเหมาะที่สุดใช้จำนวนครั้งของการหาค่าฟังก์ชันเป้าหมายเพื่อหาแนวของผลเฉลยที่ไม่ถูกครอบงำเท่ากันคือ 1250 ครั้ง จากรูปที่ 7-7 พบว่าแนวของผลเฉลยที่ได้จาก SPEA-RSM25 มีค่าดีกว่าแนวของผลเฉลยที่ได้จาก SPEA50 รูปที่ 7-8 แสดงการเปรียบเทียบแนวของผลเฉลยที่ได้จาก SPEA50 กับแนวของผลเฉลยของ SPEA-RSM01 ผลที่ได้จากการเปรียบเทียบพบว่า แนวของผลเฉลยของ SPEA-RSM01 มีค่าเกือบจะเท่ากับแนวของผลเฉลยของ SPEA50 ทั้งๆ ที่ SPEA50 ใช้การหาค่าของฟังก์ชันเป้าหมาย 1250 ครั้ง แต่ SPEA-RSM01 ใช้การหาค่าฟังก์ชันเป้าหมายแค่ 50 ครั้งเท่านั้น ซึ่งการเปรียบเทียบนี้เป็นการแสดงให้เห็นอย่างชัดเจนว่า SPEA-RSM มีสมรรถนะในการออกแบบดีกว่า SPEA

การเปรียบเทียบสมรรถนะของวิธีการออกแบบทั้งสองวิธีด้วยตัวบ่งชี้ C แสดงไว้ในรูปที่ 7-9 ซึ่งเป็นกราฟแท่งแสดงค่าตัวบ่งชี้ C เปรียบเทียบกันระหว่าง SPEA50 กับ SPEA-RSM01 SPEA-RSM05 SPEA-RSM10 SPEA-RSM15 SPEA-RSM20 และ SPEA-RSM25 ผลที่ได้จากการเปรียบเทียบในรูปที่ 7-9 สามารถอธิบายได้ดังนี้

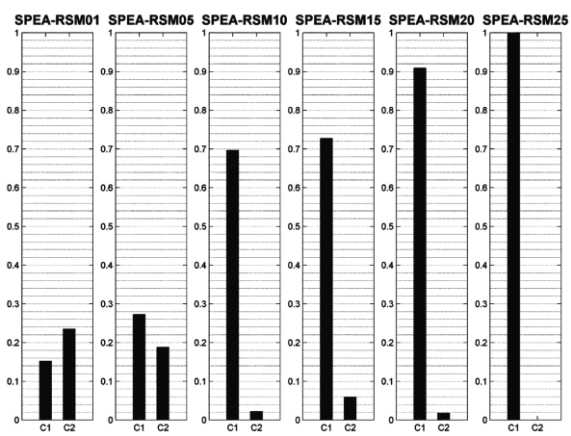
กราฟแท่งด้านซ้ายมือสุดแสดงตัวบ่งชี้ C เปรียบเทียบกันระหว่าง SPEA50 กับ SPEA-RSM01 ค่า C_1 คือ $C(\text{SPEA} - \text{RSM01}, \text{SPEA50})$ ค่า C_2 คือ $C(\text{SPEA50}, \text{SPEA} - \text{RSM01})$ ซึ่งจากผลการเปรียบเทียบพบว่าแนวของผลเฉลยของ SPEA50 ดีกว่าแนวของผลเฉลยของ SPEA-RSM01 เล็กน้อยพิจารณาได้จากค่าของ C_2 มากกว่า C_1 แต่หลังจากนั้นในการเปรียบเทียบตัวบ่งชี้ C ในรอบการคำนวณที่มากขึ้นพบว่าค่า C_1 มีค่ามากกว่าค่า C_2 ทุกกราฟ โดยเฉพาะการเปรียบเทียบกันระหว่าง SPEA-RSM25 กับ SPEA50 (กราฟแท่งสุดท้ายขวามือสุด) พบว่าแนวของผลเฉลยที่ได้จาก SPEA-RSM25 ไม่ถูกครอบงำหรือเท่ากับโดยแนวของผลเฉลยจาก SPEA50 เลย นั่นแสดงว่าเมื่อเปรียบเทียบสมรรถนะกันแล้ว SPEA-RSM จะมีสมรรถนะในการหาค่าตอบของการออกแบบดีกว่า SPEA



รูปที่ 7-7 เปรียบเทียบแนวผลเฉลยของพารेटระหว่าง SPEA50 กับ SPEA-RSM25



รูปที่ 7-8 เปรียบเทียบแนวผลเฉลยของพารेटระหว่าง SPEA50 กับ SPEA-RSM01



รูปที่ 7-9 แสดงการเปรียบเทียบโดยใช้ตัวบ่งชี้ C

แนวผลเฉลยของพารโตะจาก SPEA-RSM25 แสดงไว้ในรูปที่ 7-10 ผลเฉลยเหมาะที่สุดของพารโตะแต่ละจุดได้แสดงไว้ตามหมายเลขจุดบนกราฟ พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนที่ได้จากผลเฉลยที่ไม่ถูกรอรับจำนวน 60 ค่าของ SPEA-RSM25 แสดงไว้ในรูปที่ 7-11 และรูปที่ 7-12 ในการพิจารณาความแม่นยำของการประมาณค่าโดยแบบจำลองพื้นผิวดตอบสนองค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (RMSE: Root Mean Square Error) ของแบบจำลองพื้นผิวดตอบสนองระหว่างที่ขั้นตอนวิธีพัฒนาการหาคำตอบของการออกแบบแสดงไว้ในรูปที่ 7-13 ซึ่งค่าความผิดพลาดเฉลี่ยคำนวณได้จากการเปรียบเทียบความแตกต่างระหว่างค่าฟังก์ชันเป้าหมายที่ได้จากการประมาณค่าของแบบจำลองพื้นผิวดตอบสนองกับค่าที่ได้จากฟังก์ชันเป้าหมายจริงค่าความคลาดเคลื่อนกำลังสองเฉลี่ยของฟังก์ชันเป้าหมายที่หนึ่งเป็นไปตามสมการดังนี้

$$RMSE_1 = \sqrt{\frac{1}{N_A} \sum_{i=1}^{N_A} (f_1(y_i) - g_1(y_i))^2} \quad (7.3)$$

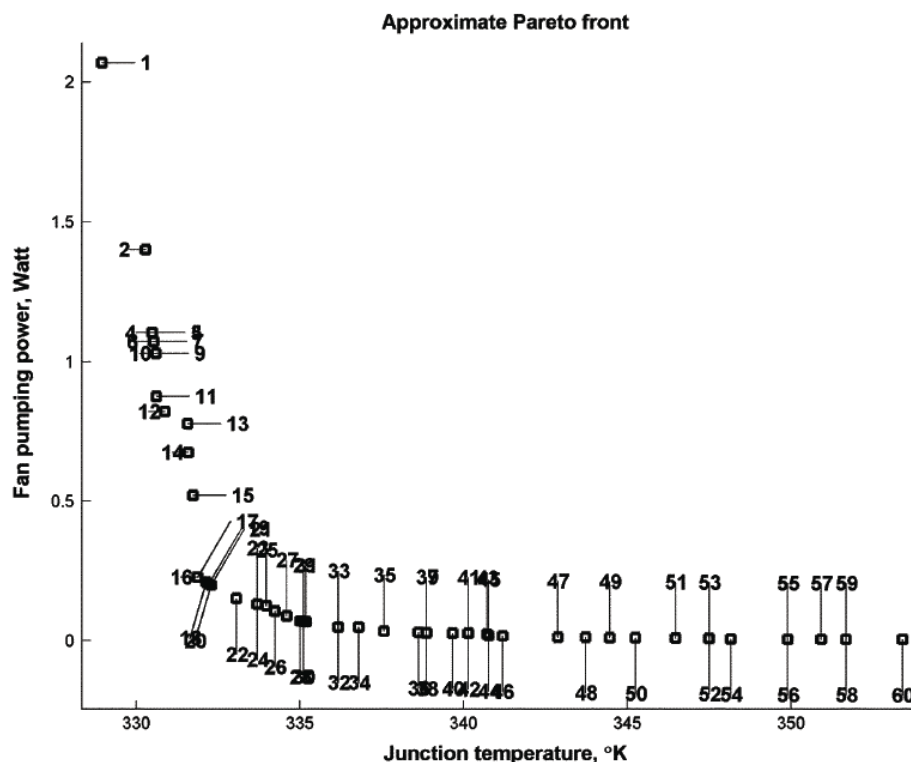
ค่าความคลาดเคลื่อนกำลังสองเฉลี่ยของฟังก์ชันเป้าหมายที่สองเป็นไปตามสมการดังนี้

$$RMSE_2 = \sqrt{\frac{1}{N_A} \sum_{i=1}^{N_A} (f_2(y_i) - g_2(y_i))^2} \quad (7.4)$$

โดยที่ y_i คือ สมาชิกเมตริกซ์เก็บผลเฉลยภายนอกของพารโตะในส่วนของแบบจำลองพื้นที่ผิวดตอบสนอง B^{t+1}

N_A คือ จำนวนสมาชิกทั้งหมดของ B^{t+1}

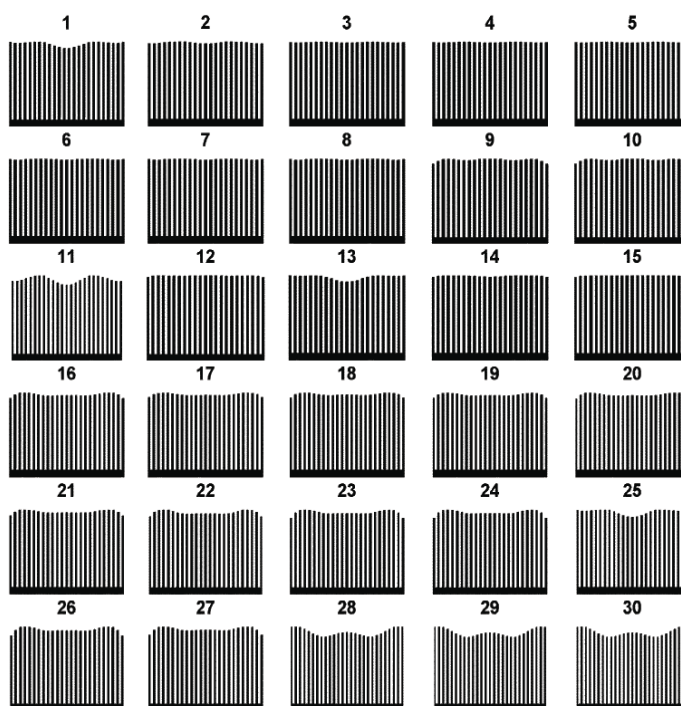
รูปที่ 7-13 แสดงค่าความคลาดเคลื่อนกำลังสองเฉลี่ยของอุณหภูมิจุดต่อระหว่างอุปกรณ์ระบายความร้อนและหน่วยประมวลผลกลางซึ่งพบว่ามีค่าความคลาดเคลื่อนสูงเมื่อเริ่มการคำนวณในช่วงแรก แต่มีแนวโน้มลดลงเมื่อรอบการคำนวณเพิ่มขึ้น ซึ่งค่าอุณหภูมิจุดต่อจากการออกแบบที่ได้อยู่ระหว่าง 330–355 °K ค่าความผิดพลาดจากการประมาณค่าอยู่ระหว่าง 10–20 °K ซึ่งถือว่าเป็นค่าความคลาดเคลื่อนที่น้อยมากและสามารถยอมรับได้ ค่าความคลาดเคลื่อนกำลังสองเฉลี่ยของกำลังที่ต้องการในการเป่าอากาศของพัดลมระบายความร้อน, f_2 , พบว่ามีค่าสูงมากในช่วงเริ่มต้นการคำนวณและมีค่าลดลงจนเกือบจะไม่มีค่าความคลาดเคลื่อนเมื่อรอบการคำนวณเพิ่มมากขึ้น ดังนั้น จากผลการศึกษาที่ได้ในรูปที่ 7-7 ถึงรูปที่ 7-9 สามารถสรุปได้ว่า RSM จะมีความแม่นยำต่ำในช่วงของการเริ่มต้นการคำนวณ ซึ่งเป็นผลให้แนวของผลเฉลยของ SPEA50 ดีกว่าแนวของผลเฉลย SPEA-RSM01 อย่างไรก็ตามเมื่อรอบของการคำนวณเพิ่มขึ้นความแม่นยำของ RSM จะมีค่าเพิ่มขึ้นจนส่งผลให้แนวของผลเฉลยของ SPEA-RSM ดีกว่า SPEA



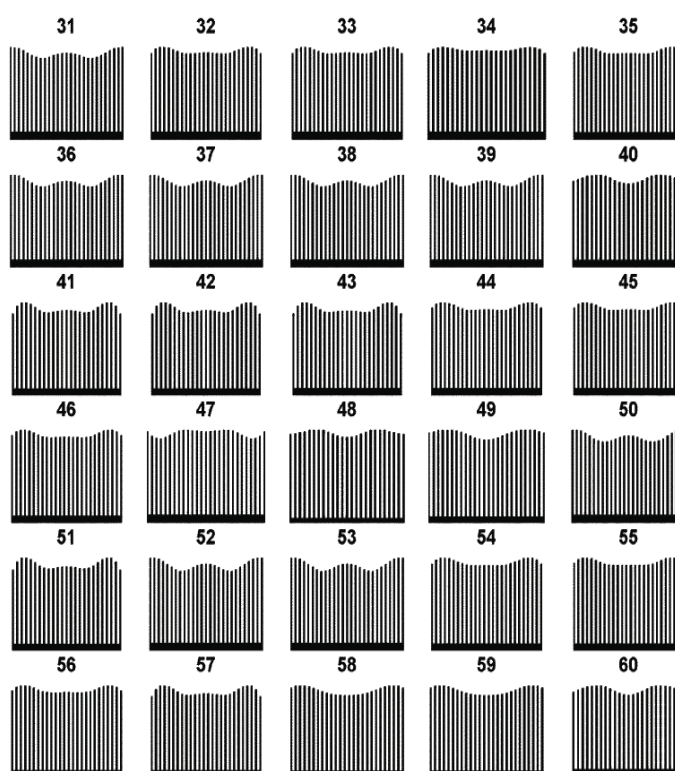
รูปที่ 7-10 แสดงแนวผลเฉลยของพารेटอของ SPEA-RSM25

7.3 สรุปผลการศึกษา

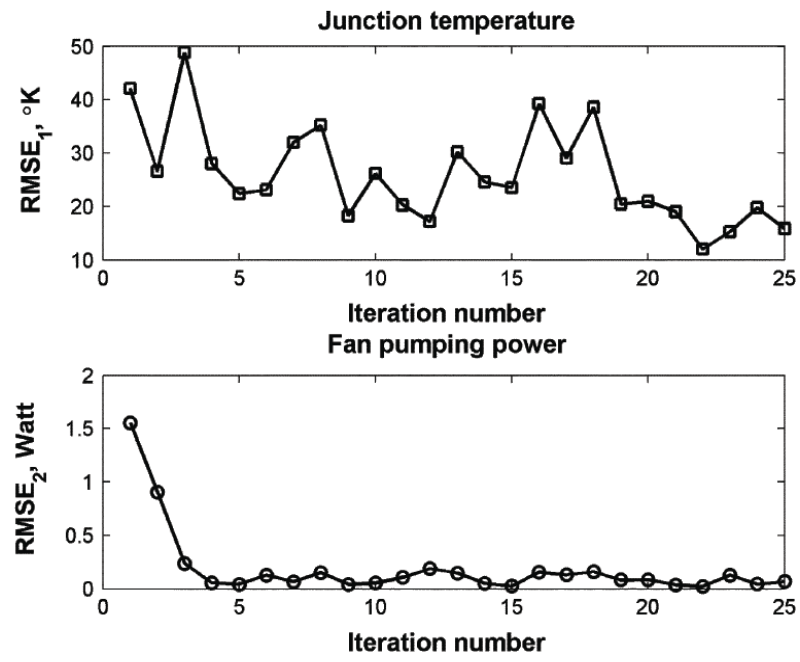
จากผลการศึกษาการออกแบบหาค่าเหมาะที่สุดด้วยขั้นตอนวิวัฒนาการแบบหลายเป้าหมายโดยใช้ SPEA-RSM เปรียบเทียบกับ SPEA ในการออกแบบอุปกรณ์ระบายความร้อนด้วยอากาศแบบครีบบ้าน ซึ่งจากการเปรียบเทียบผลที่ได้ของทั้งสองวิธีพบว่าแนวผลเฉลยของพารेटอที่ได้จาก SPEA-RSM ดีกว่าแนวผลเฉลยที่ได้จาก SPEA แม้จำนวนรอบของการดำเนินการของ SPEA-RSM จะมีค่าน้อยกว่าก็ตาม อัตราการหาค่าตอบของ SPEA-RSM เร็วกว่าของ SPEA พิจารณาได้จากการเปรียบเทียบผลไว้ตามรูปที่ 7-8 และรูปที่ 7-9 ดังนั้นจึงสามารถสรุปได้ว่าการออกแบบหาค่าเหมาะที่สุดแบบหลายเป้าหมายด้วย SPEA-RSM เป็นวิธีการหาผลเฉลยของปัญหาที่ดีและมีสมรรถนะในการหาค่าตอบสูง สามารถนำไปออกแบบสำหรับปัญหาที่มีฟังก์ชันเป้าหมายซับซ้อนและต้องใช้เวลาในการวิเคราะห์คำตอบเพื่อลดเวลาในการออกแบบได้จริง



รูปที่ 7-11 แสดงรูปร่างพื้นที่หน้าตัดที่ได้จาก SPEA-RSM25 ลำดับที่ 1- 30



รูปที่ 7-12 แสดงรูปร่างพื้นที่หน้าตัดที่ได้จาก SPEA-RSM25 ลำดับที่ 31- 60



รูปที่ 7-13 แสดงค่าความคลาดเคลื่อนกำลังสองเฉลี่ยของฟังก์ชันเป้าหมาย

การออกแบบหาค่าเหมาะที่สุดแบบหลายเป้าหมาย ของอุปกรณ์ระบายความร้อนแบบครีบกวน

8.1 ความนำ

การศึกษาในบทนี้เป็นการศึกษาการออกแบบอุปกรณ์ระบายความร้อนต่อเนื่องจากการศึกษาในบทที่ 6 และบทที่ 7 ที่ผ่านมา โดยได้ทำการประยุกต์ใช้ PBIL ร่วมกับโปรแกรมพลศาสตร์ของไหลเชิงคำนวณในการออกแบบหาค่าเหมาะที่สุดของอุปกรณ์ระบายความร้อนแบบครีบกวน สำหรับการศึกษาในบทนี้ได้ทำการศึกษาเปรียบเทียบสมรรถนะในการออกแบบของขั้นตอนวิธีวิวัฒนาการ PBIL กับขั้นตอนวิธีวิวัฒนาการอีก 2 ขั้นตอนวิธี คือ MPSO และ SPEA ทั้งนี้ในการศึกษาได้เพิ่มความซับซ้อนของรูปร่างอุปกรณ์ระบายความร้อนเพื่อให้สามารถออกแบบรูปร่างอุปกรณ์ระบายความร้อนให้มีสมรรถนะในการระบายความร้อนได้ดีขึ้น โดยแบ่งการศึกษาการออกแบบอุปกรณ์ระบายความร้อนออกเป็น 3 กรณีดังนี้

1. อุปกรณ์ระบายความร้อนแบบครีบกวนที่มีความสูงของครีบกวนสามารถปรับเปลี่ยนได้
 2. อุปกรณ์ระบายความร้อนแบบครีบกวนที่มีความสูงของครีบกวนที่แต่สามารถปรับเปลี่ยนมุมเอียงได้
 3. อุปกรณ์ระบายความร้อนแบบครีบกวนที่มีความสูงและมุมเอียงของครีบกวนสามารถปรับเปลี่ยนได้
- ผลการศึกษาในแต่ละกรณีของขั้นตอนวิธีวิวัฒนาการทั้ง 3 ขั้นตอนวิธีข้างต้นจะถูกนำมาเปรียบเทียบกันเพื่อหาผลลัพธ์ของการออกแบบที่ดีที่สุด รวมทั้งเปรียบเทียบผลลัพธ์ที่ได้จากแต่ละขั้นตอนวิธีเพื่อหาขั้นตอนวิธีวิวัฒนาการที่เหมาะสมสำหรับปัญหาการออกแบบอุปกรณ์ระบายความร้อน รายละเอียดผลการศึกษานำเสนอไว้ดังต่อไปนี้

8.2 รูปร่างของอุปกรณ์และตัวแปรออกแบบ

สำหรับการออกแบบหาค่าเหมาะที่สุดของอุปกรณ์ระบายความร้อนแบบครีบกวนโดยใช้วิธีวิวัฒนาการแบบสองเป้าหมายของการศึกษาครั้งนี้ ได้กำหนดให้มีการออกแบบอุปกรณ์ระบายความร้อน 3 แบบ และได้แบ่งประเภทของตัวแปรออกแบบเป็น 3 กลุ่ม ดังนี้

F1 : อุปกรณ์ระบายความร้อนที่มีความสูงของครีบกวนสามารถปรับเปลี่ยนได้

F2 : อุปกรณ์ระบายความร้อนที่มีความสูงของครีบกวนที่แต่ปรับเปลี่ยนมุมเอียงได้

F3 : อุปกรณ์ระบายความร้อนที่มีความสูงและมุมเอียงของครีบกวนสามารถปรับเปลี่ยนได้

ตัวแปรออกแบบกลุ่ม F1 ใช้ตัวแปรออกแบบเช่นเดียวกับการศึกษาในบทที่ 7 โดยมีจำนวนของตัวแปรออกแบบทั้งหมด 8 ตัวแปรออกแบบ

ตัวแปรออกแบบกลุ่ม F2 ใช้ในการออกแบบอุปกรณ์ระบายความร้อนแบบครีบริดตรงเอียงโดยที่ความสูงของครีบริดระบายความร้อนมีค่าคงที่ ซึ่งข้อดีของอุปกรณ์ระบายความร้อนประเภทนี้คือการส่งผ่านความร้อนจากอุปกรณ์อิเล็กทรอนิกส์ดีกว่าอุปกรณ์ระบายความร้อนที่มีครีบริดตรงในแนวดิ่งเนื่องจากมีพื้นที่ผิวในการระบายความร้อนมากกว่าในขณะที่พื้นที่ฐานและความสูงของอุปกรณ์ระบายความร้อนมีค่าเท่ากัน ในกระบวนการหาค่าเหมาะที่สุดของอุปกรณ์ระบายความร้อนกลุ่มตัวแปร F2 ให้มีการเปลี่ยนแปลงรูปร่างอัตโนมัติ นั้น จำเป็นจะต้องใช้ระเบียบวิธีคำนวณเชิงตัวเลขที่มีความซับซ้อนมาประยุกต์ใช้เนื่องจากจะมีความยุ่งยากในส่วนของกริดในการสร้างกริดในโปรแกรมพลศาสตร์ของไหลเชิงคำนวณ ถ้าครีบริดระบายความร้อนมีการเอียงเข้ามาชนกันหรือใกล้กันมากเกินไป เมื่อเกิดกรณีแบบนี้จึงจำเป็นต้องมีการปรับให้ระยะห่างและความสูงของครีบริดให้เหมาะสม โดยใช้เงื่อนไขการปรับตามสมการที่ (8.1) เพื่อให้แน่ใจว่าช่องว่างระหว่างครีบริดระบายความร้อนมีความกว้างมากพอที่จะให้อากาศไหลผ่านเข้าไประบายความร้อนได้ รูปร่างของครีบริดเอียง ตัวแปร X_i และ X_{i+1} ของอุปกรณ์ระบายความร้อนแสดงไว้ในรูปที่ 8-1

$$X_{i+1} \geq X_i \quad (8.1)$$

เวกเตอร์ของตัวแปรออกแบบของชุดนี้มีจำนวนสมาชิกทั้งหมด 8 ตัวแปร ดังรายละเอียดต่อไปนี้

$x_1 = d$, ความหนาของครีบริดระบายความร้อน

$x_2 = n$, จำนวนของครีบริดระบายความร้อน

$x_3 = t$, ความหนาของฐานอุปกรณ์ระบายความร้อน

$x_4 = S_1$, ตัวแปรที่ใช้ในการควบคุมและกำหนดค่า X_i ของครีบริดระบายความร้อน

$x_5 = S_2$, ตัวแปรที่ใช้ในการควบคุมและกำหนดค่า X_i ของครีบริดระบายความร้อน

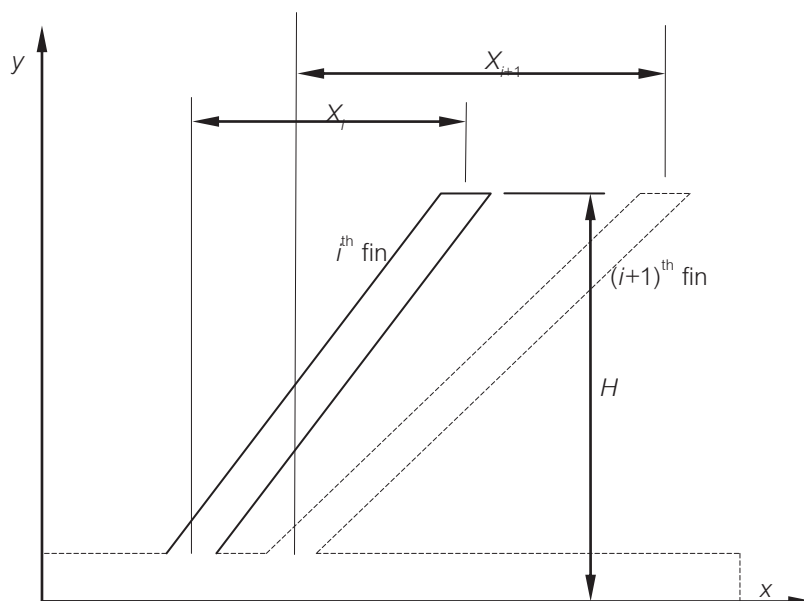
$x_6 = S_3$, ตัวแปรที่ใช้ในการควบคุมและกำหนดค่า X_i ของครีบริดระบายความร้อน

$x_7 = S_4$, ตัวแปรที่ใช้ในการควบคุมและกำหนดค่า X_i ของครีบริดระบายความร้อน

$x_8 = V_f$, ความเร็วของอากาศเข้าอุปกรณ์ระบายความร้อน

โดยที่ในการออกแบบของกลุ่มตัวแปร F2 ยังคงกำหนดให้พื้นที่ฐานมีขนาดเดียวกันกับตัวแปรออกแบบในกลุ่ม F1 คือความกว้างของอุปกรณ์ระบายความร้อน (W) และความยาวของอุปกรณ์ระบายความร้อน (L) มีขนาดเท่ากับ 60 มิลลิเมตร และ 80 มิลลิเมตร ตามลำดับ ในขณะที่ความสูงของครีบริดระบายความร้อน H กำหนดให้มีขนาดคงที่เท่ากับ 40 มิลลิเมตร จากเงื่อนไขในสมการที่ (8.1) ทำให้เกิดข้อจำกัดในการออกแบบข้อใหม่ขึ้นมา ซึ่งมีความสัมพันธ์ตามสมการที่ (8.2) ดังนี้

$$S_1 \leq S_2 \leq S_3 \leq S_4. \quad (8.2)$$



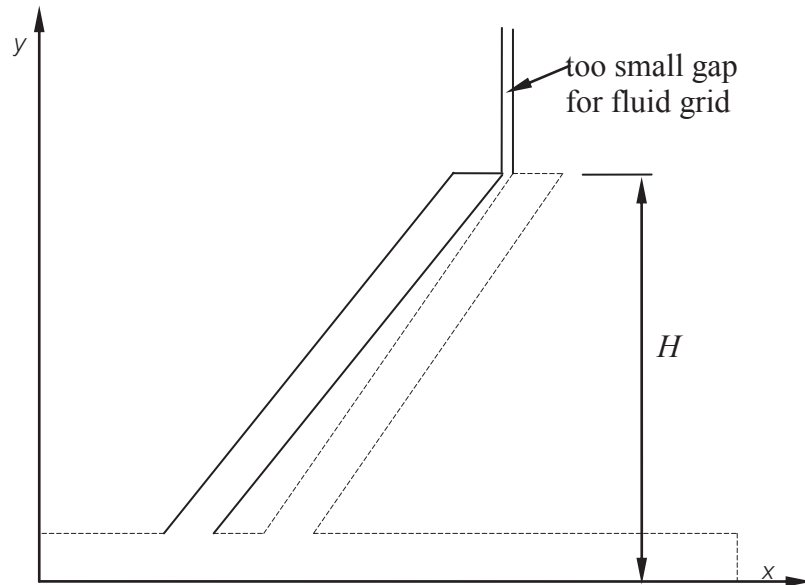
รูปที่ 8-1 แสดงครีบเอียงของอุปกรณ์ระบายความร้อน

ข้อจำกัดตามสมการที่ (8.2) ก่อให้เกิดความยุ่งยากในกระบวนการหาค่าที่เหมาะสมที่สุดด้วยPBILของการศึกษาครั้งนี้ ดังนั้นเพื่อหลีกเลี่ยงปัญหาจากข้อจำกัดข้างต้น ค่าของตัวแปร S_1 , S_2 , S_3 และ S_4 ซึ่งเป็นตัวแปรที่ใช้ในการควบคุมความเอียงของครีบระบายความร้อนจึงถูกแก้ไขให้มีค่าตามสมการที่ (8.3)

$$\begin{aligned} s_1 &= \frac{S_1}{S_1 + S_2 + S_3 + S_4} S_4 \\ s_2 &= \frac{S_1 + S_2}{S_1 + S_2 + S_3 + S_4} S_4 \\ s_3 &= \frac{S_1 + S_2 + S_3}{S_1 + S_2 + S_3 + S_4} S_4 \\ s_4 &= S_4 \end{aligned} \quad (8.3)$$

จากการใช้สมการที่ (8.3) ทำให้มั่นใจได้ว่าข้อจำกัดของการออกแบบตามสมการที่ (8.2) สามารถเป็นจริงได้ ขอบเขตล่างและขอบเขตบนของตัวแปรออกแบบที่ใช้ในการควบคุมการกระจายความเอียงของครีบ ถูกกำหนดค่าให้เป็นดังนี้ S_1 , S_2 , S_3 มีค่าอยู่ระหว่าง 0-100 มิลลิเมตร และ S_4 มีค่าอยู่ระหว่าง 0-20 มิลลิเมตร

ปัญหาของการสร้างกริดในโปรแกรมพลศาสตร์ของไหลเชิงคำนวณเกิดขึ้นเนื่องจากช่องว่างระหว่างครีบที่ i^{th} กับครีบที่ $(i+1)^{th}$ มีขนาดเล็กมากดังแสดงไว้ในรูปที่ 8-2 ซึ่งสามารถแก้ไขได้ด้วยการลดขนาดความสูงของครีบที่ i^{th} ดังแสดงไว้ในรูปที่ 8-3 นั้นหมายความว่าในกรณีในการออกแบบของกลุ่มตัวแปร F2 มีความเป็นไปได้ว่าความสูงของครีบระบายความร้อนจะมีขนาดไม่เท่ากันกับค่าที่กำหนดไว้ให้คงที่



รูปที่ 8-2 แสดงรูปร่างของครีบริบที่มีช่องว่างระหว่างครีบบีบแบนเกินไป

ขั้นตอนการคำนวณและการหาคำตอบของการออกแบบกลุ่มตัวแปร F2 สามารถเขียนได้ดังต่อไปนี้

- 1: ป้อนค่าเมตริกซ์เลขฐานสองของตัวแปรออกแบบทั้งหมด B
- 2: ถอดรหัสเมตริกซ์ของตัวแปรออกแบบ B ให้เป็นเวกเตอร์ของตัวแปรออกแบบที่เป็นจำนวน

จริง x

- 3: กำหนดค่า $d = x_1$, $n = \text{round}(x_2)$, $t = x_3$, $S_1 = x_4$, $S_2 = x_5$, $S_3 = x_6$, $S_4 = x_7$, $V_f = x_8$

- 4: หาค่าระยะห่างระหว่างครีบริบระบายความร้อนโดยใช้สมการดังนี้

$$b = \frac{W - nd}{n - 1}$$

- 5: ถ้าระยะห่างระหว่างครีบริบ b แคบหรือกว้างมาก ($b \notin [0.5, 5] \text{ mm}$) ให้เพิ่มหรือลดค่าจำนวนครีบริบจนกว่าระยะห่างระหว่างครีบริบ b จะเข้ามาอยู่ในช่วงที่กำหนด

- 6: หาค่า s_1, s_2, s_3 และ s_4 จากสมการที่ (8.3)

- 7: ประมาณค่ากลางของตัวแปรอิสระ $\left\{ \frac{W}{7}, \frac{3W}{7}, \frac{5W}{7}, \frac{W}{2} \right\}$ และตัวแปรตาม $\{S_1, S_2, S_3, S_4\}$ โดย

ใช้การประมาณค่ากลางแบบ cubic spline

- 8: หาค่า $X_i, i = 1, \dots, n$

- 9: คัดความสูงของครีบริบระบายความร้อน สำหรับการสร้างกริดของโปรแกรมพลศาสตร์ของไหล

เชิงคำนวณ

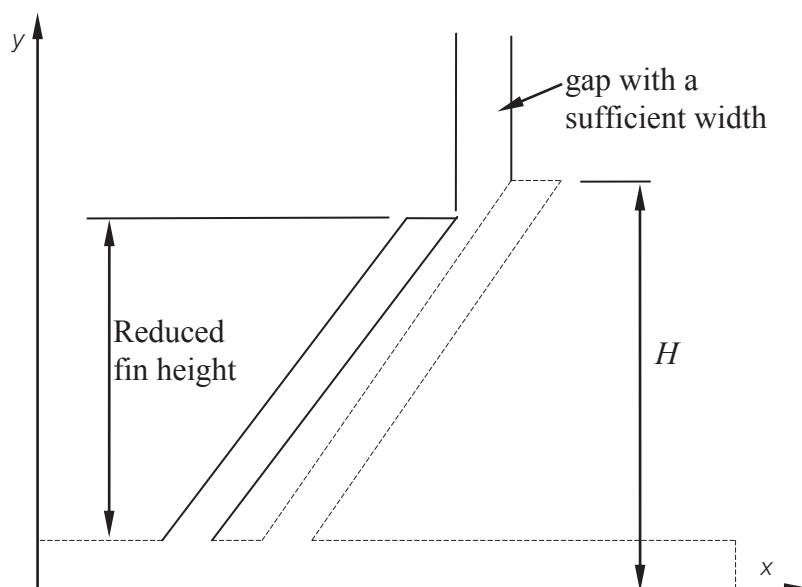
- 9.1 For $j = (n - 1)$ to the middle fin

- 9.2 ถ้าช่องว่างระหว่างครีบริบที่ j^{th} กับครีบริบที่ $(j + 1)^{\text{th}}$ แคบมาก ให้ลดความสูงของครีบริบที่ j^{th}

ลงจนกว่าช่องว่างระหว่างครีบริบจะกว้างเพียงพอ

9.3 Next j

- 10: สร้างความสูงและความเอียงของครีกระบายความร้อน
- 11: สร้างพื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนโดยใช้ตัวแปรทั้งหมดจากที่กล่าวมาข้างต้น
- 12: สร้างอินพุตไฟล์สำหรับการวิเคราะห์ด้วยโปรแกรมพลศาสตร์ของไหลเชิงคำนวณ
- 13: นำไปวิเคราะห์ด้วยโปรแกรมพลศาสตร์ของไหลเชิงคำนวณ
- 14: อ่านค่าเอาต์พุตที่ได้จากโปรแกรมและนำไปคำนวณค่าฟังก์ชันเป้าหมาย f_1 และ f_2 เพื่อนำไปป้อนให้กับกระบวนการหาค่าเหมาะที่สุดต่อไป



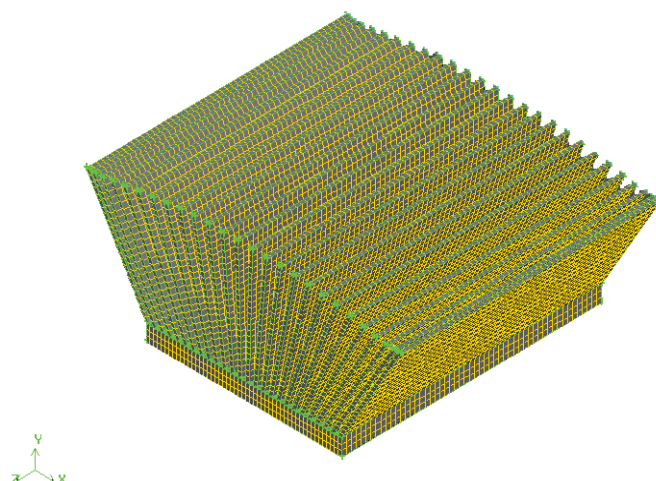
รูปที่ 8-3 แสดงการลดความสูงของครีบลงเพื่อให้ช่องว่างระหว่างครีบมีความเหมาะสม

ตัวแปรออกแบบกลุ่ม F3 ใช้สำหรับการออกแบบอุปกรณ์ระบายความร้อนแบบครีบตรงเอียงโดยที่ความสูงและมุมเอียงของครีกระบายความร้อนสามารถเปลี่ยนแปลงค่าได้ เป็นการรวมเอาตัวแปรออกแบบของกลุ่ม F1 และกลุ่ม F2 เข้าด้วยกัน ตัวแปรออกแบบที่ใช้ในการออกแบบมีทั้งหมด 12 ตัวแปร ประกอบด้วยตัวแปรต่าง ๆ ดังนี้

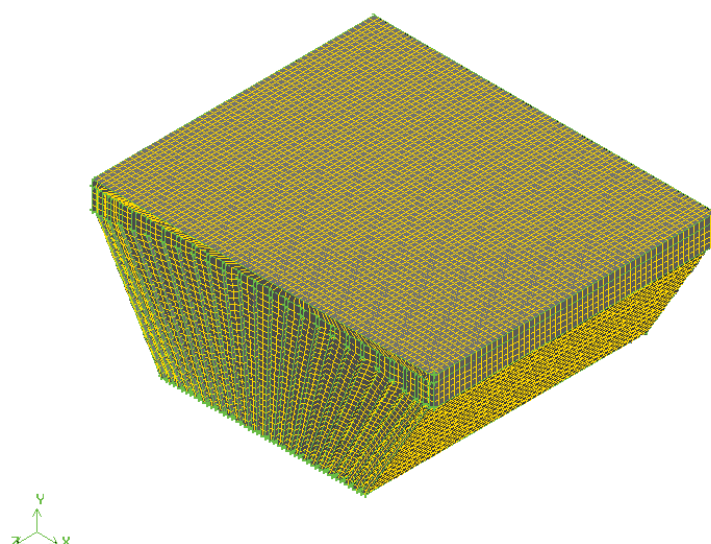
$$x = [d, n, t_b, V_f, H_1, H_2, H_3, H_4, S_1, S_2, S_3, S_4]$$

ขั้นตอนการคำนวณและการหาคำตอบของฟังก์ชันเป้าหมายของกลุ่มตัวแปร F3 นี้คล้ายกับขั้นตอนการคำนวณของกลุ่มตัวแปรออกแบบ F2 เพียงแต่เพิ่มการประมาณค่ากลางของการกระจายตัวของความสูงของครีกระบายความร้อนเข้าไปด้วยเท่านั้น

แบบจำลองของแข็งของอุปกรณ์ระบายความร้อนที่มีครีบตรงเอียง ซึ่งถูกแบ่งเป็นกริดสำหรับใช้ในการวิเคราะห์ด้วยวิธีปริมาตรจำกัดนำเสนอไว้ในรูปที่ 8-4 สำหรับกริดของแบบจำลองของของไหลนำเสนอไว้ในรูปที่ 8-5



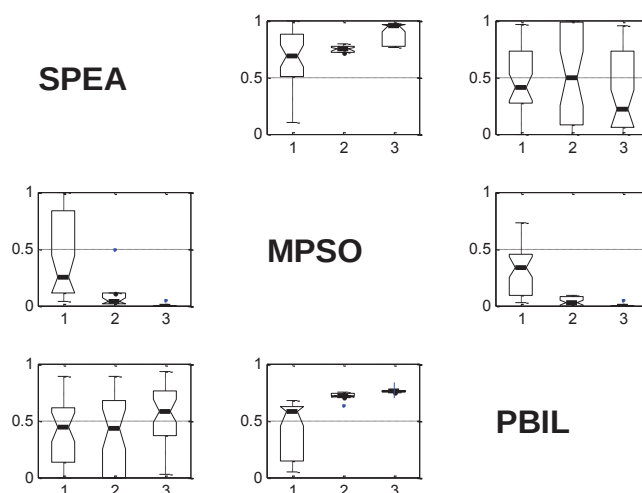
รูปที่ 8-4 แบบจำลองของแข็งสำหรับการวิเคราะห์ด้วยวิธีปริมาตรจำกัดของครีบเสียง



รูปที่ 8-5 แบบจำลองของของไหลสำหรับการวิเคราะห์ด้วยวิธีปริมาตรจำกัด

8.3 ผลการศึกษา

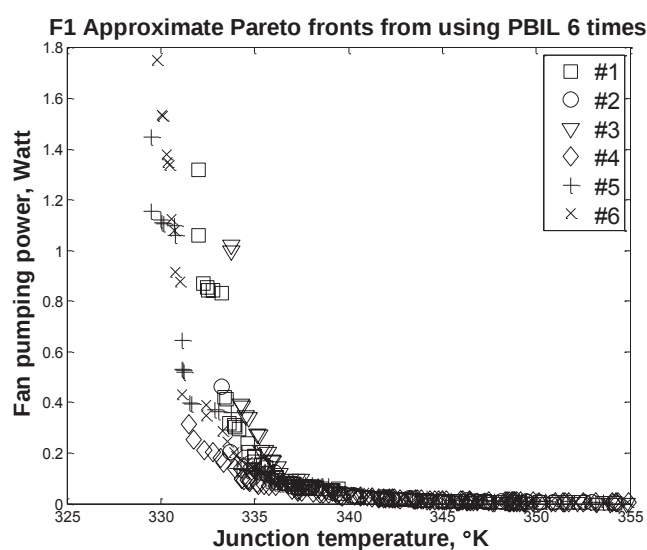
แนวผลเฉลยที่ไม่ถูกรอบงำทั้งหมดจำนวน $3 \times 3 \times 6$ ผลเฉลยจากการดำเนินการ 6 ครั้งของวิธีวิวัฒนาการแบบหลายฟังก์ชันเป้าหมาย 3 ขั้นตอนวิธีที่ถูกใช้ในการแก้ปัญหาการออกแบบ 3 กลุ่มตัวแปรออกแบบ เมื่อนำผลเฉลยที่ได้จากการออกแบบด้วยขั้นตอนวิธีวิวัฒนาการทั้งสามมาทำการเปรียบเทียบสมรรถนะโดยใช้พารามิเตอร์บ่งชี้สมรรถนะคือตัวบ่งชี้ C (C-comparison) ผลการเปรียบเทียบกันของตัวบ่งชี้สมรรถนะ C ของการหาค่าเหมาะที่สุดแสดงไว้ในรูปที่ 8-6 ดังนี้



รูปที่ 8-6 แสดงการเปรียบเทียบสมรรถนะของวิธีพัฒนาการโดยใช้ตัวบ่งชี้สมรรถนะ C

จากรูปที่ 8-6 พบว่าSPEAและPBILมีสมรรถนะสูงกว่าMPSO โดยที่ SPEAและPBILมีสมรรถนะเท่ากันในกรณีของกลุ่มตัวแปรออกแบบ F1 และกลุ่มตัวแปรออกแบบ F2 (SPEAมีสมรรถนะดีกว่าPBILเล็กน้อยในกรณีของกลุ่มตัวแปรออกแบบ F2) สำหรับกลุ่มตัวแปรออกแบบ F3พบว่าPBILมีสมรรถนะดีกว่า SPEA ซึ่งทำให้เราได้ข้อสรุปของการศึกษาเปรียบเทียบครั้งนี้ว่าขั้นตอนวิธีพัฒนาการที่ให้ผลโดยภาพรวมดีที่สุดคือPBIL ดังนั้นในการนำเสนอต่อจากนี้จะนำเสนอเฉพาะผลที่ได้จากการใช้PBILในหาคำตอบเท่านั้น

แนวของพารेटที่ได้จากการแก้ปัญหาของกลุ่ม F1 จำนวน 6 ครั้งแสดงไว้ในรูปที่ 8-7 ซึ่งจะพบว่าผลที่ได้จากPBILจากการดำเนินการทั้งหมด 6 ครั้ง มีค่าแทบจะสอดคล้องกันทุกครั้ง จะเห็นได้จากแนวของพารेटเกือบจะเป็นแนวเดียวกันทั้ง 6 ครั้ง

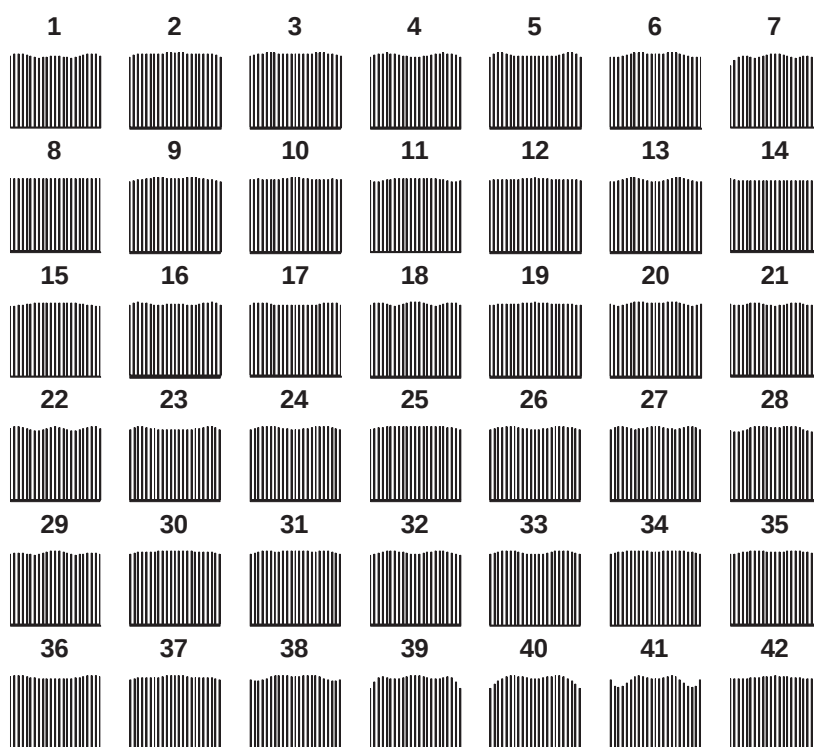


รูปที่ 8-7 ประมาณแนวของพารेट 6 ครั้งของกลุ่มตัวแปรออกแบบ F1

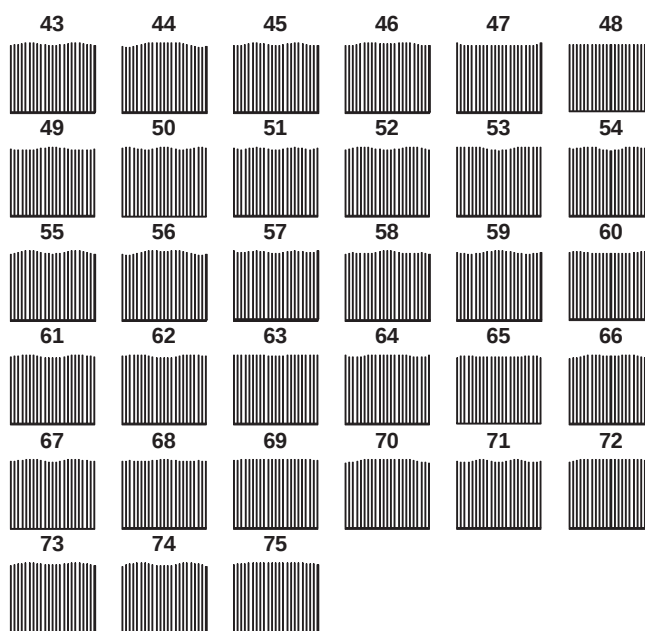
พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนได้จากแนวพารโอดีที่สุด (ถูกเลือกโดยตัวบ่งชี้สถานะ C) ของกลุ่มตัวแปรออกแบบ F1 แสดงไว้ในรูปที่ 8-8 และรูปที่ 8-9 จากภาพดังกล่าวพบว่าอุปกรณ์ระบายความร้อนทั้งหมดมีแนวโน้มที่จะมีจำนวนครีบเท่ากันและอุปกรณ์ระบายความร้อนทั้งหมดมีความสูงของครีบต่างกันเล็กน้อยเท่านั้น ในขณะที่เดียวกันมีอุปกรณ์ระบายความร้อนเพียงบางส่วนเท่านั้นที่มีการเปลี่ยนแปลงความสูงของครีบ

แนวของพารโอดีที่ได้จากการแก้ปัญหากลุ่มตัวแปรออกแบบ F2 จำนวน 6 ครั้งแสดงไว้ในรูปที่ 8-10 ซึ่งพบว่าสมรรถนะในการหาผลเฉลยของPBILสำหรับกรณีกลุ่มตัวแปรออกแบบ F2 นี้ยังสามารถพิจารณาได้ว่ามีความสอดคล้องกันอยู่ทั้ง 6 ครั้งของการดำเนินการหาผลเฉลย ถึงแม้ว่าแนวของพารโอดีทั้งหมดจะไม่ได้เป็นแนวเดียวกันก็ตาม ทั้งนี้พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนที่ได้จากแนวผลเฉลยของพารโอดีที่ดีที่สุดของกรณีนี้นำเสนอไว้ในรูปที่ 8-11 และรูปที่ 8-12

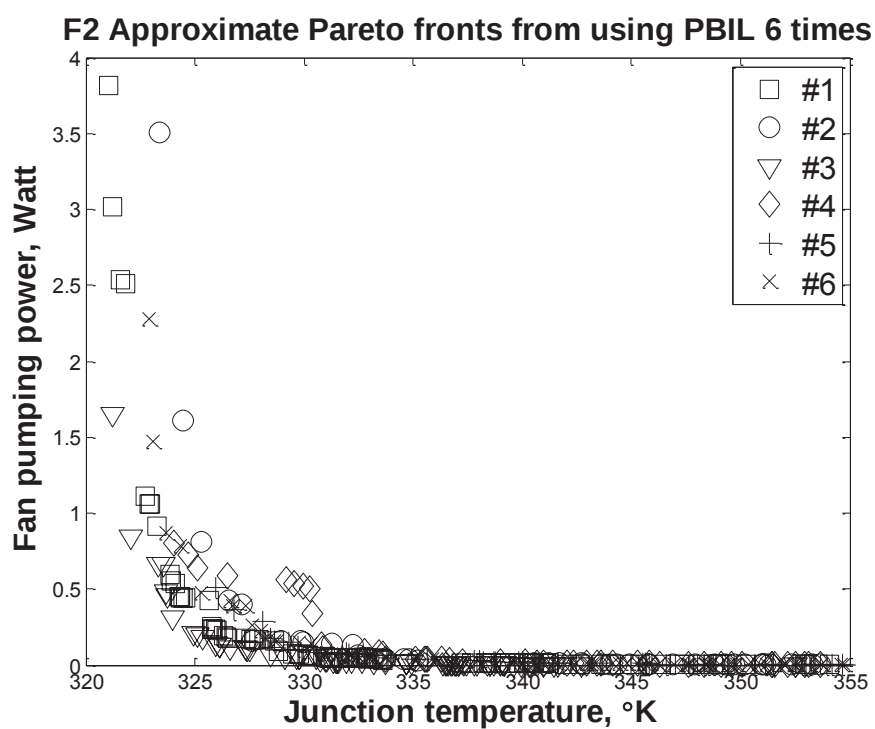
แนวของพารโอดีที่ได้จากการแก้ปัญหากลุ่มตัวแปรออกแบบ F3 จำนวน 6 ครั้งแสดงไว้ในรูปที่ 8-13 ซึ่งพบว่าสมรรถนะในการหาผลเฉลยของPBILสำหรับกรณีกลุ่มตัวแปรออกแบบ F3 นี้มีความสอดคล้องกันน้อยที่สุดเมื่อเปรียบเทียบกับปัญหาที่ผ่านมาทั้งสองปัญหา พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนที่ได้จากแนวผลเฉลยของพารโอดีที่ดีที่สุดของกรณี F3 นำเสนอไว้ในรูปที่ 8-14 และรูปที่ 8-15



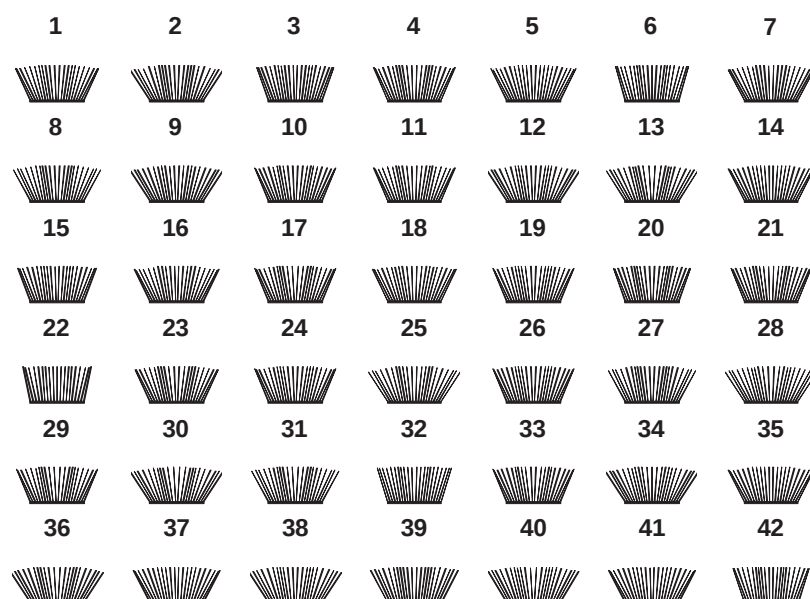
รูปที่ 8-8 พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนจากแนวผลเฉลยที่ดีที่สุดของ F1 : I



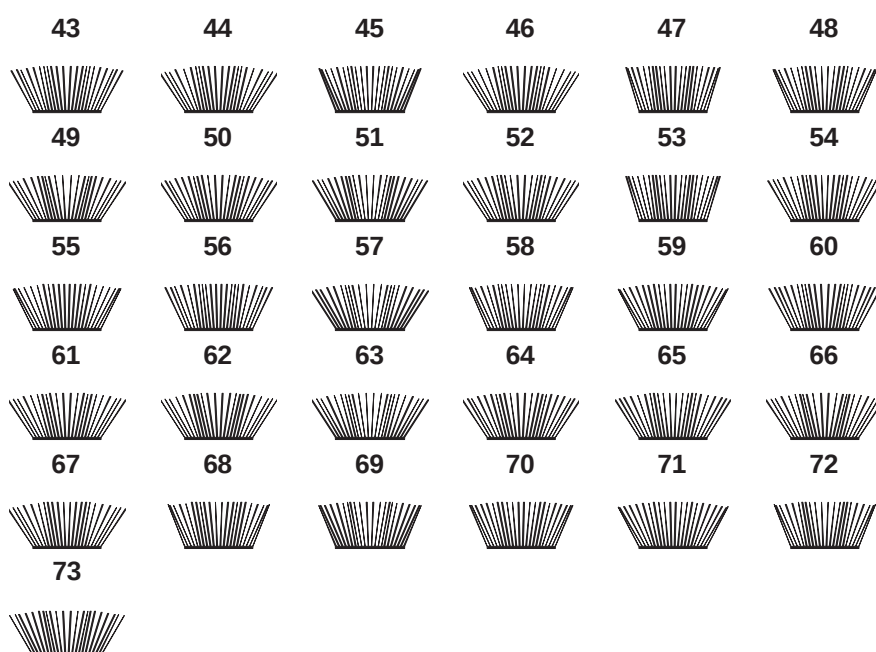
รูปที่ 8-9 พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนจากแนวผลเฉลยที่ดีที่สุดของ F1 : II



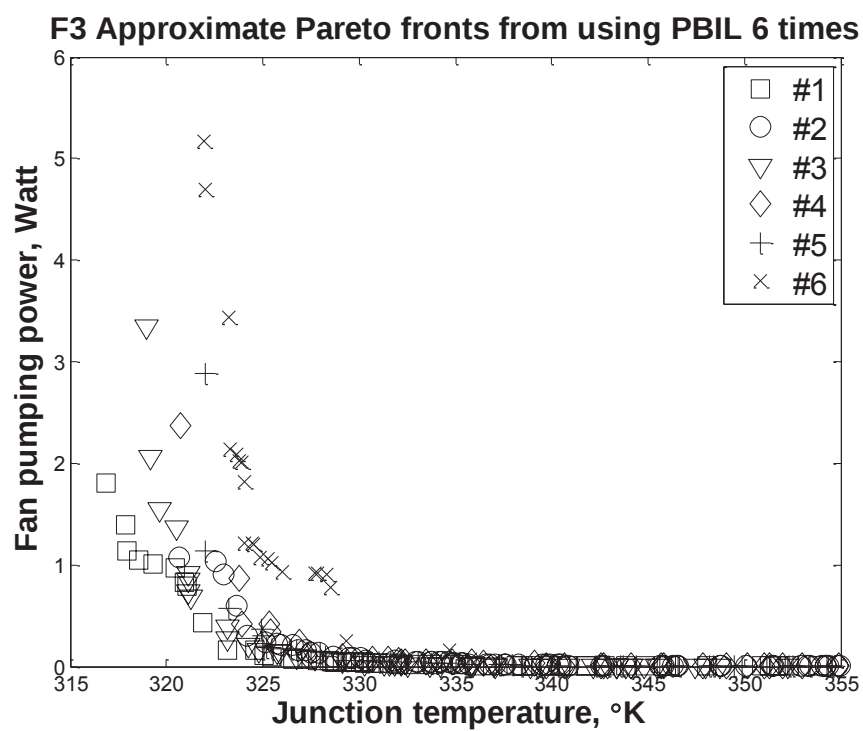
รูปที่ 8-10 ประมาณแนวผลเฉลยของพารेटอ 6 ครั้งของกลุ่มตัวแปรออกแบบ F2



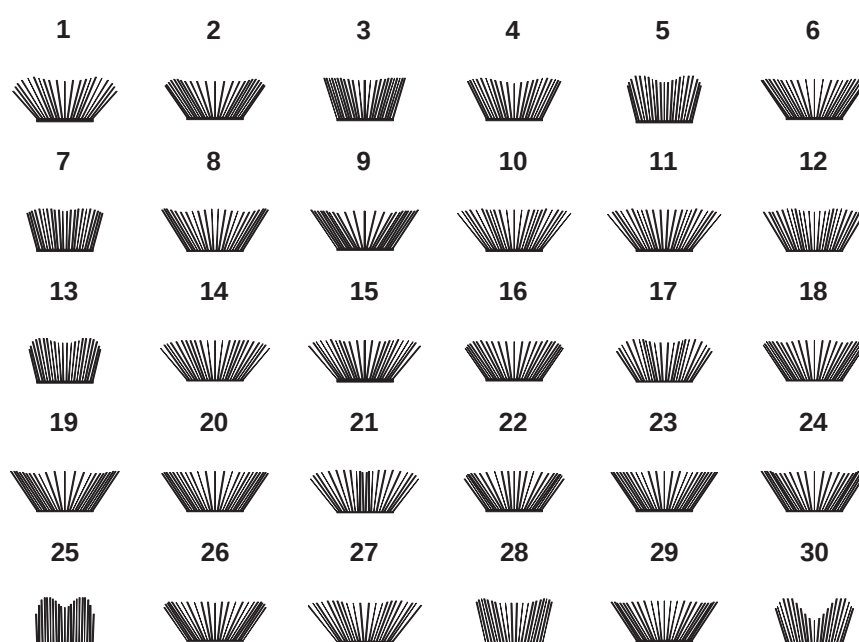
รูปที่ 8-11 พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนจากแนวผลเฉลยที่ดีที่สุดของ F2 : I



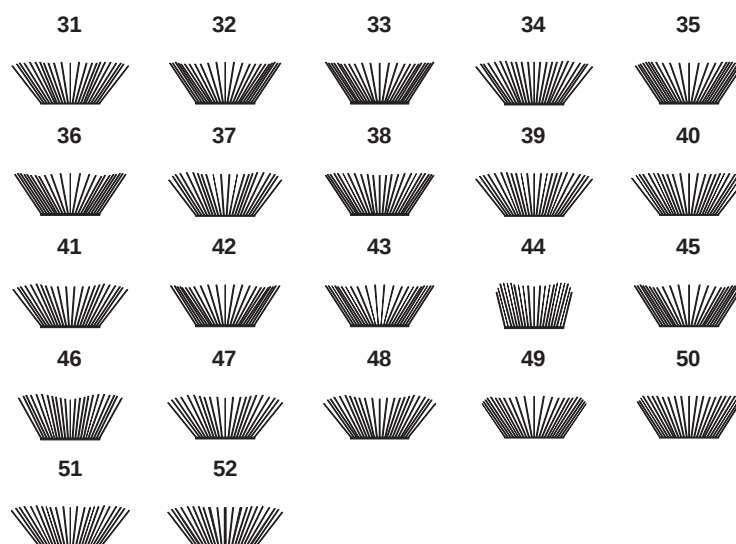
รูปที่ 8-12 พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนจากแนวผลเฉลยที่ดีที่สุดของ F2 : II



รูปที่ 8-13 ประมาณแนวผลเฉลยของพารามิเตอร์จำนวน 6 ครั้งของตัวแปรออกแบบ F3

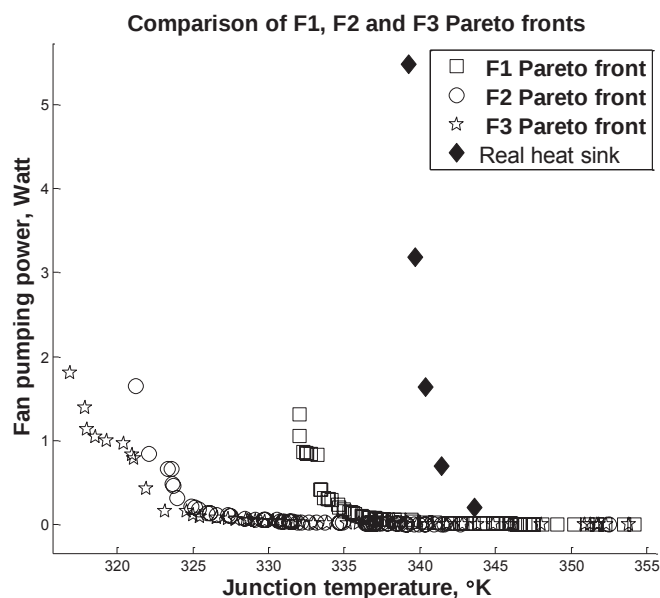


รูปที่ 8-14 พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนจากแนวผลเฉลยที่ดีที่สุดของ F3 : I



รูปที่ 8-15 พื้นที่หน้าตัดของอุปกรณ์ระบายความร้อนจากแนวผลเฉลยที่ดีที่สุดของ F3 : II

แนวของผลเฉลยพารโตที่ดีที่สุดของปัญหาการออกแบบ F1, F2, และ F3 แสดงเปรียบเทียบกันไว้ในรูปที่ 8-16 ดังนี้



รูปที่ 8-16 เปรียบเทียบแนวผลเฉลยของพารโตที่ดีที่สุดของปัญหา F1, F2 และ F3

จากรูปที่ 8-16 เครื่องหมายที่เป็นรูปเพชร ($\diamond$) แสดงถึงค่าฟังก์ชันเป้าหมายที่ได้จากการวิเคราะห์อุปกรณ์ระบายความร้อนที่มีใช้จริงในท้องตลาดด้วยโปรแกรมพลศาสตร์ของไหลเชิงคำนวณ (รูปร่างของอุปกรณ์ระบายความร้อนแสดงไว้ในรูปที่ 8-17) โดยที่ค่าตัวแปรต่าง ๆ ของอุปกรณ์ระบายความร้อนเป็นดังนี้

$V_f \in \{2,3,4,5,6\}$ เมตรต่อวินาที, $t=8$ มิลลิเมตร, $d=2$ มิลลิเมตร, $H=32$ มิลลิเมตร, $n=21$, $W=63$ มิลลิเมตร และ $L=80$ มิลลิเมตร ซึ่งจากรูปที่ 8-16 พบว่าแนวผลเฉลยของพารามิเตอร์ที่ดีที่สุดของปัญหา F1 และ F2 ถูกครอบงำโดยแนวผลเฉลยของพารามิเตอร์ของปัญหา F3 ซึ่งหมายความว่าแนวผลเฉลยของปัญหา กลุ่มตัวแปรออกแบบ F3 ดีที่สุด ผลเฉลยที่ให้ค่าผลเฉลยรองลงมาก็คือผลเฉลยของ F2 และผลเฉลยของ F1 ตามลำดับ นั้นแสดงว่าการออกแบบโดยใช้ตัวแปรออกแบบที่มีความหลากหลายอย่างเช่นปัญหา F3 ทำให้ได้ อุปกรณ์ระบายความร้อนที่ดีกว่า



รูปที่ 8-17 อุปกรณ์ระบายความร้อนที่มีใช้งานจริง

8.4 สรุปผลการศึกษา

จากผลการศึกษาพบว่าขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายโดยประยุกต์ใช้PBILในการหาคำตอบหรือผลเฉลยของปัญหาการหาค่าเหมาะที่สุดสามารถออกแบบรูปร่างของอุปกรณ์ระบายความร้อนได้ดี และจากการที่แบ่งการศึกษาออกเป็น 3 กลุ่มตัวแปรออกแบบเพื่อเปรียบเทียบกัน ซึ่งผลที่ได้คือผลเฉลยจากกลุ่มตัวแปรออกแบบ F3 ดีกว่าผลเฉลยจากกลุ่มตัวแปรออกแบบ F2 และผลเฉลยจากกลุ่มตัวแปรออกแบบ F2 ดีกว่าผลเฉลยของกลุ่มตัวแปรออกแบบ F1 นั้น ทำให้ทราบว่าในการออกแบบหาค่าเหมาะที่สุดการใช้กลุ่มตัวแปรที่มีความหลากหลายจะทำให้ได้รูปร่างของอุปกรณ์ที่ดีกว่ากลุ่มตัวแปรที่มีจำกัด เมื่อพิจารณาผลเฉลยที่ได้จากวิธีวิวัฒนาการโดยใช้PBILจากทั้ง 3 ปัญหาพบที่มีความสอดคล้องกัน นอกจากนี้จากการทำการศึกษาเปรียบเทียบกันระหว่างMPSO SPEA และPBILพบว่าจากสมรรถนะของแต่ละวิธีในการศึกษา 3 ปัญหาการออกแบบ PBILมีสมรรถนะในการหาคำตอบดีที่สุดเป็นอันดับที่หนึ่ง รองลงมาก็คือ SPEA และ MPSO ตามลำดับ

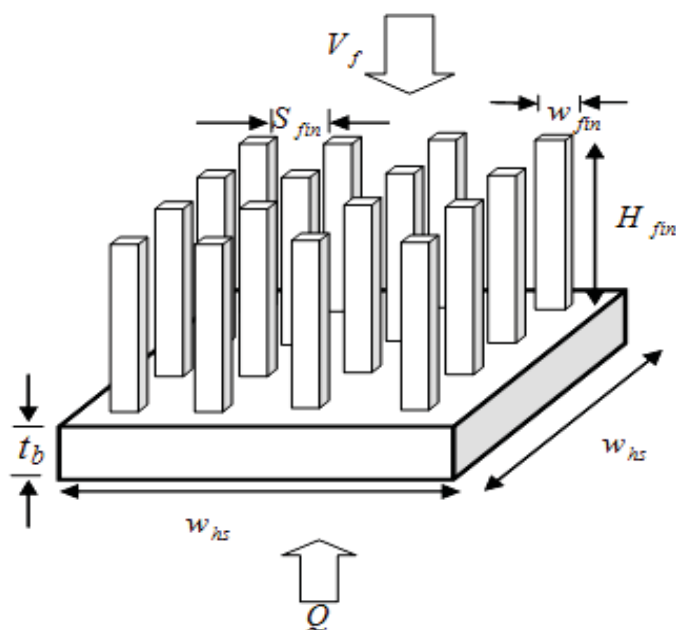
การออกแบบอุปกรณ์ระบายความร้อนแบบครีบท่่ง

9.1 ความนำ

เนื้อหาในบทนี้ต่อเนื่องจากบทที่ 6-8 เป็นการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายกับการออกแบบหาค่าเหมาะที่สุดของอุปกรณ์ระบายความร้อนแบบครีบท่่ง ครีบท่่งในที่นี้คือ ครีบท่่งหน้าตัดสี่เหลี่ยมจัตุรัส แบบจำลองทางคณิตศาสตร์เพื่ออธิบายการไหลและการถ่ายเทความร้อนในอุปกรณ์ รวมถึงตัวแปรที่ใช้ จะมีรูปแบบเดียวกันกับเนื้อหาก่อนหน้านี้ ปัญหาการออกแบบหลายเป้าหมายก็เป็นปัญหาการออกแบบชนิดเดียวกันกับที่นำเสนอในบทที่ 7-8 ในบทนี้ จะนำเสนอการออกแบบอุปกรณ์ระบายความร้อนแบบครีบท่่งที่มีครีบท่่งตรงและเอียง จากนั้นจะเป็นการนำเสนอการประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการที่ใช้แบบจำลองทดแทนในการเพิ่มสมรรถนะในการหาผลเฉลยของกระบวนการออกแบบอุปกรณ์ระบายความร้อนแบบครีบท่่ง เนื้อหาในบทนี้จะรวบรัดและสั้นกว่าบทที่ผ่านมา เพราะได้อธิบายรายละเอียดของแบบจำลองทางคณิตศาสตร์และปัญหาการออกแบบแล้ว

9.2 การออกแบบอุปกรณ์ระบายความร้อนแบบครีบท่่งตรง

รูปที่ 9-1 แสดงแบบจำลองของอุปกรณ์ระบายความร้อนแบบครีบท่่งตรง ตัวแปรที่เพิ่มขึ้นมา นอกเหนือจากอุปกรณ์แบบครีบท่่งแผ่นคือขนาดความกว้างของแท่งครีบท w_{fin} และช่องว่างระหว่างแท่งครีบท s_{fin}



รูปที่ 9-1 อุปกรณ์ระบายความร้อนแบบครีบท่่งตรง

9.2.1 การเข้ารหัสและถอดรหัสตัวแปรออกแบบ

กระบวนการสำคัญในการออกแบบรูปร่างของอุปกรณ์ระบายความร้อน ด้วยขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายคือ การเข้ารหัสและถอดรหัสตัวแปรออกแบบ การเข้ารหัสคือการกำหนดตัวแปรออกแบบว่าจะใช้ตัวแปรเลขฐานสอง เลขจำนวนเต็ม หรือเลขจำนวนจริง ส่วนการถอดรหัสคือการแปรผลจากตัวแปรออกแบบไปเป็นขนาดและรูปทรงของอุปกรณ์ เพื่อสร้างแบบจำลองสำหรับวิเคราะห์เชิงตัวเลขต่อไป ในการศึกษาี้ ขนาดของเวกเตอร์ออกแบบคือ 21×1 หรือมีตัวแปรกำหนดขนาดและรูปทรงของอุปกรณ์ระบายความร้อนแบบครีบบางตรงทั้งสิ้น 21 ตัวแปร ซึ่งประกอบด้วย จำนวนครีบบาง ขนาดฐานของอุปกรณ์ ความหนาของฐาน ตัวแปรสำหรับระบุความสูงของแท่งครีบบาง 16 ค่า และขนาดของครีบบางหน้าตัดสี่เหลี่ยม กระบวนการถอดรหัส สามารถทำได้ ดังต่อไปนี้

HeatSinkGeometry

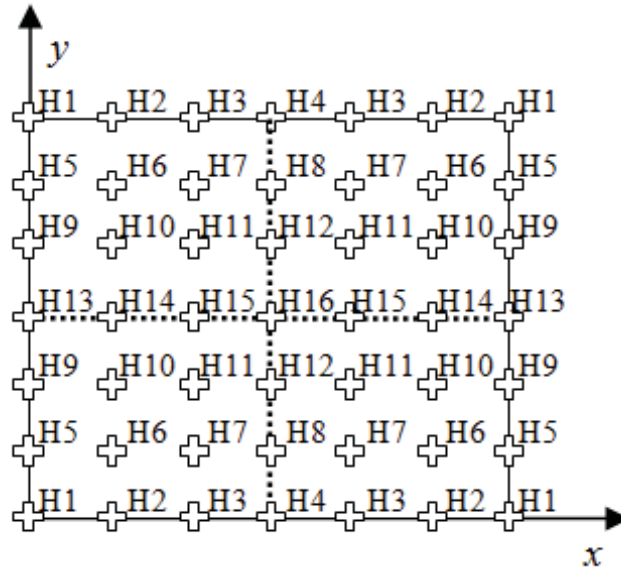
ตัวแปรขาเข้า $\mathbf{x}$ ขนาด 21×1

ข้อมูลขาออก ขนาดและรูปร่างของอุปกรณ์ระบายความร้อน

- 1: กำหนดค่าพารามิเตอร์ต่างๆ
 $W_{fin} = x_1, N_f = \text{round}(x_2), t_b = x_3, \mathbf{H}_0 = \{H_1, \dots, H_{16}\} = \{x_4, \dots, x_{19}\}, V_f = x_{20}, w_{hs} = x_{21}$
- 2: คำนวณค่าของ $s_{fin} = (w_{hs} - N_f \cdot w_{fin}) / (N_f - 1)$
- 3: ถ้าเงื่อนไขคือ $S_{fin} > 0.004$ ดำเนินการ
 - 3.1: $S_{fin} = 0.004$
 - 3.2: $N_f = \text{round}((w_{hs} + S_{fin}) / (w_{fin} + S_{fin}))$
- 4: หรือถ้าเงื่อนไขคือ $s_{fin} < 0.0025$ คำนวณค่าใหม่ของ N_f เป็น
 - 4.1: $N_f = \text{round}((w_{hs} + 0.0025) / (w_{fin} + 0.0025))$
 - 4.2: $s_{fin} = (w_{hs} - N_f \cdot w_{fin}) / (N_f - 1)$
- จบเงื่อนไขตรรกศาสตร์
- 5: คำนวณสัมประสิทธิ์ของการประมาณค่าในช่วง $\mathbf{c} = \text{SurfSpline}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{H}_0)$
- 6: หาค่าหนึ่งของครีบบางแบบแท่งทั้งหมด $\mathbf{x}_1, \mathbf{y}_1$
- 7: หาค่าความสูงที่ตำแหน่งต่างๆของครีบบาง $\mathbf{H}_1 = \text{SplineEval}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{c}, \mathbf{x}_0, \mathbf{y}_0)$
- 8: บันทึกข้อมูลขนาดและรูปร่างของอุปกรณ์ระบายความร้อน เพื่อการวิเคราะห์ทางไฟไนต์เอลิเมนต์

N_f คือจำนวนของครีบบางในแต่ละแถว ฟังก์ชัน $\text{round}(N_f)$ คือการปัดเศษค่า N_f ให้เป็นจำนวนเต็ม ส่วนฟังก์ชัน $\text{SurfSpline}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{H}_0)$ คือการสร้างแบบจำลองอินเตอร์โพลชันโดยใช้ฟังก์ชันฐานหลักแนวรัศมีเมื่อตัวแปรต้นคือ $\mathbf{x}_0$, และ $\mathbf{y}_0$ ส่วนตัวแปรตามคือ $\mathbf{H}_0$ ผลที่ได้จากอินเตอร์โพลชันคือเมตริกซ์สัมประสิทธิ์ $\mathbf{c}$ ฟังก์ชัน $\text{SplineEval}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{c}, \mathbf{x}_0, \mathbf{y}_0)$ คือการทำนายค่าความสูงของครีบบางที่จุด $\mathbf{x}_1, \mathbf{y}_1$ เมื่อทำนายครบทุกค่า จะได้การกระจายตัวของความสูงของครีบบางแบบสมมาตร

รูปที่ 9-2 แสดงตำแหน่งของจุดควบคุมเพื่อคำนวณความสูงของแท่งครีป จากรูปเป็นตำแหน่งที่มีความสูงตามค่าของเอลิเมนต์ของ $\mathbf{H}_0$ ที่กระจายกันอยู่อย่างสมมาตร 49 จุด พิกัดของจุดเหล่านี้คือตัวแปร $\mathbf{x}_0$, และ $\mathbf{y}_0$ ในกระบวนการถอดรหัสตัวแปรออกแบบ (จุดที่เขียนด้วยสัญลักษณ์ +)



รูปที่ 9-2 ตำแหน่งของจุดควบคุม

เมื่อทราบค่าของ $\mathbf{H}_0$, $\mathbf{x}_0$ และ $\mathbf{y}_0$ เวกเตอร์สัมประสิทธิ์ของอินเตอร์โพลชันแบบใช้ฟังก์ชันฐานหลักแนวลรีมี $\mathbf{c}$ สามารถหาได้จากผลเฉลยของระบบสมการเชิงเส้นดังนี้

$$\sum_{j=1}^{49} K(d_{ij})c_j = H_i \text{ for } i = 1, \dots, 49 \quad (9.1)$$

เมื่อ $d_{ij} = \sqrt{(x_{0,i} - x_{0,j})^2 + (y_{0,i} - y_{0,j})^2}$

$K(d_{ij}) =$ ฟังก์ชันฐานหลักแนวลรีมี (ในที่นี้ $K(d_{ij}) = d_{ij}$) และ

H_i คือความสูงของครีปที่จุดควบคุม $(x_{0,i}, y_{0,i})$ ดังแสดงในรูปที่ 9-3

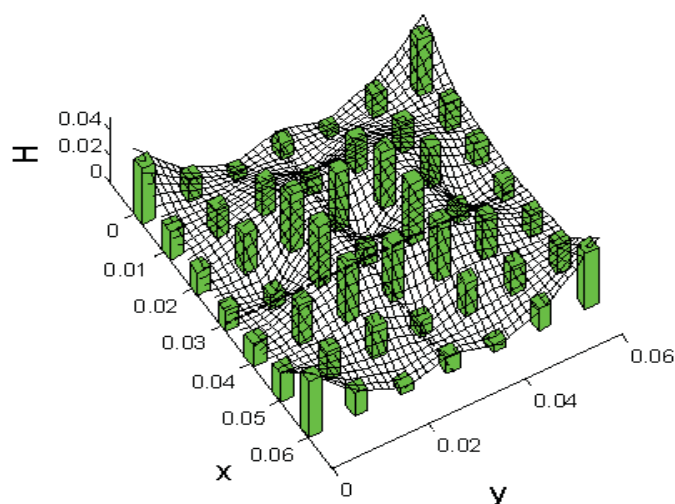
กระบวนการนี้คือขั้นตอนที่ 5 $\mathbf{c} = \text{SurfSpline}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{H}_0)$ ในการถอดรหัสตัวแปร

หลังจากระบุพิกัดในการสร้างแท่งครีปทั้งหมด N_f^2 แท่งจะได้พิกัด $\mathbf{x}_1$ และ $\mathbf{y}_1$ ความสูงของแท่งครีปที่ตำแหน่งของจุดเหล่านี้ สามารถคำนวณได้จาก

$$H_{1,i} = \sum_{j=1}^{49} K(d_{ij})c_j \quad (9.2)$$

เมื่อ $d_{ij} = \sqrt{(x_{1,i} - x_{0,j})^2 + (y_{1,i} - y_{0,j})^2}$

ซึ่งก็คือขั้นตอนที่ 7 $\mathbf{H}_1 = \text{SplineEval}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{c}, \mathbf{x}_0, \mathbf{y}_0)$ ในการถอดรหัสตัวแปรออกแบบ รูปที่ 9-3 แสดงพื้นผิวที่ได้จากการอินเตอร์โพลชัน จากสมการ (9.1)



รูปที่ 9-3 การกระจายความสูงของครีบ

9.2.2 ผลการรันขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย

เมื่อได้กระบวนการถอดรหัสที่เหมาะสมแล้ว เราสามารถดำเนินการหาค่าฟังก์ชันในการออกแบบ โดยการวิเคราะห์ทางไฟไนต์เอลิเมนต์ และสามารถประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการที่ใช้เลขฐานสองหรือจำนวนจริงก็ได้ในการหาผลเฉลย ขั้นตอนวิธีวิวัฒนาการที่เลือกใช้เพื่อหาผลเฉลยของปัญหาการออกแบบที่เหมาะสมที่สุดแบบหลายเป้าหมาย (ปัญหาเดียวกันกับบทที่ 7 และ 8) ของอุปกรณ์ระบายความร้อนแบบครีบแท่งตรง คือ PBIL, SPEA, PSO, และ AMOSA [6] ทำการรันแต่ละขั้นตอนวิธีวิวัฒนาการเพื่อหาคำตอบของปัญหาการออกแบบ 3 รอบ โดยใช้ขนาดประชากรเป็น 25 และจำนวนวงวนสูงสุดคือ 35 รอบ ซึ่งหมายความว่าแต่ละวิธีใช้จำนวนครั้งในการวิเคราะห์ไฟไนต์เอลิเมนต์ $35 \times 25 = 875$ ครั้ง ซึ่งใช้เวลาในการคำนวณมาก ขนาดของเมตริกซ์เก็บผลเฉลยที่ไม่ถูกครอบงำคือ 50

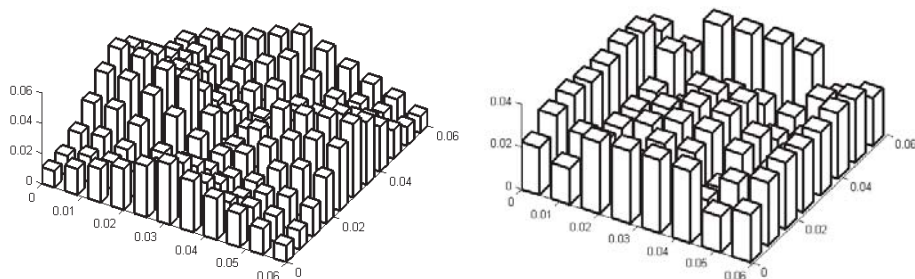
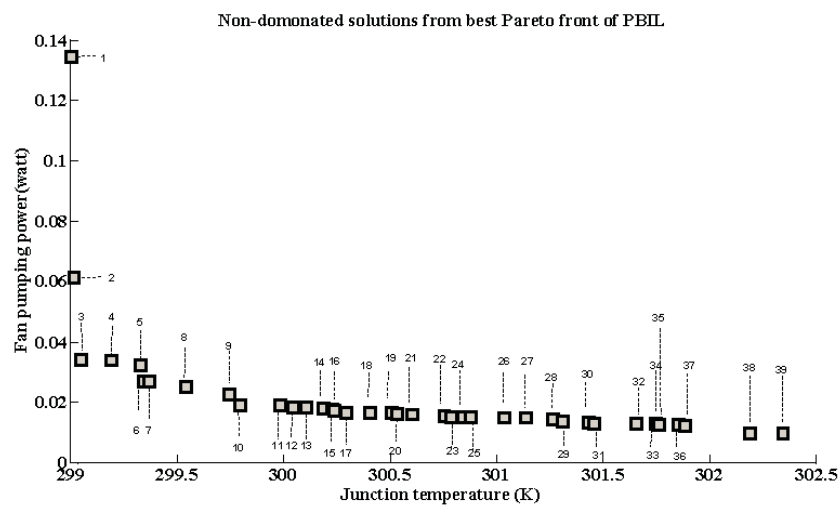
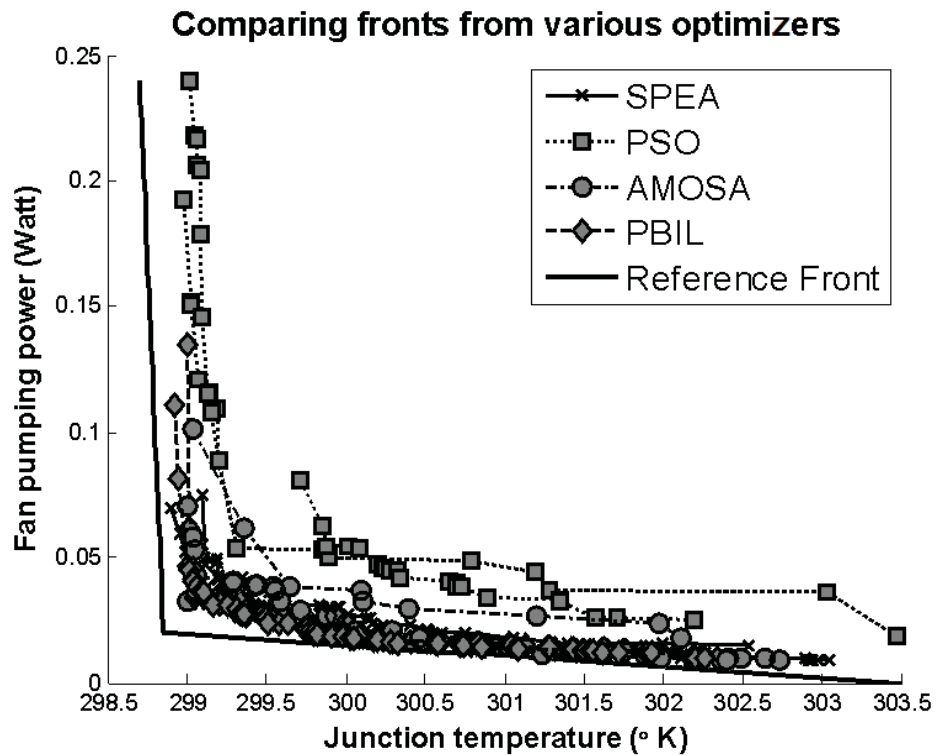
ผลการรันแต่ละวิธีวิวัฒนาการแบบหลายเป้าหมายในการออกแบบอุปกรณ์นี้ 3 ครั้ง แสดงในรูปที่ 9-4 ในแต่ละรอบของการรันขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย ใช้เวลาประมาณ 43.75-58.33 ชั่วโมง การเปรียบเทียบค่า HV และค่า generational distance (GD) ของแนวพาเรโตในรูปที่ 9-4 แสดงในตารางที่ 9-1 จากผลการเปรียบเทียบพบว่า วิธีวิวัฒนาการแบบหลายเป้าหมายที่ดีที่สุดสำหรับปัญหาการออกแบบนี้ คือ PBIL รองลงมาคือ SPEA2 ส่วนวิธีที่ให้ผลการออกแบบเป็นอันดับสุดท้ายคือ PSO

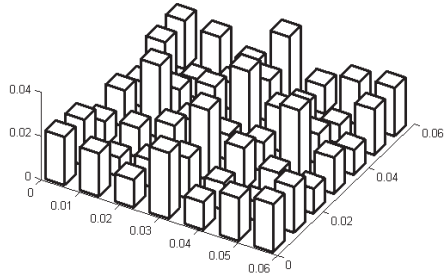
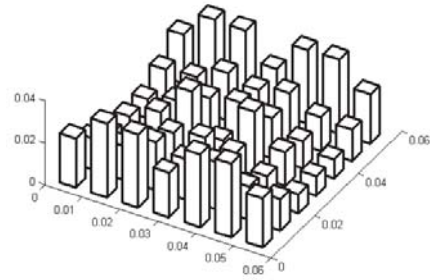
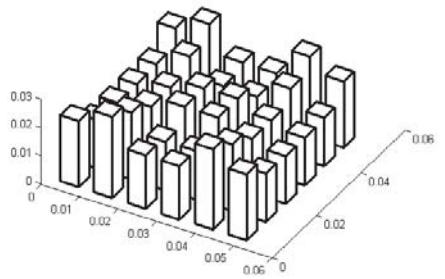
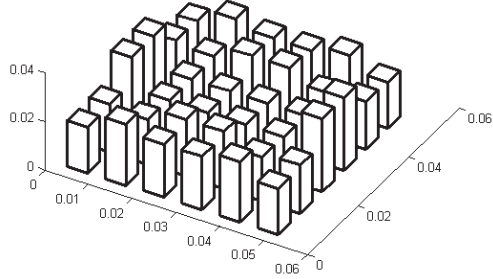
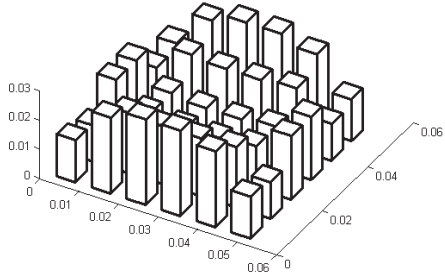
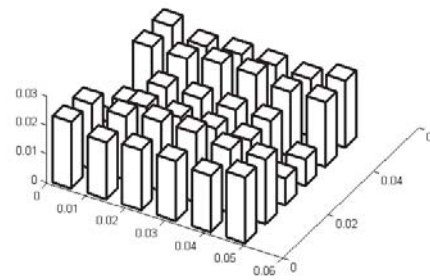
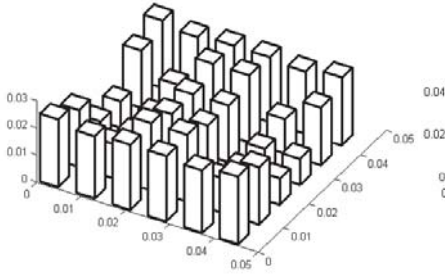
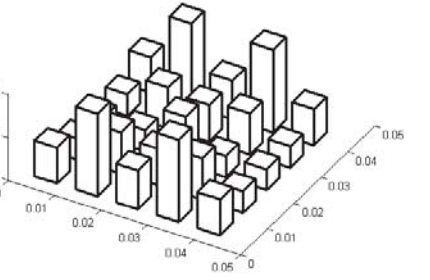
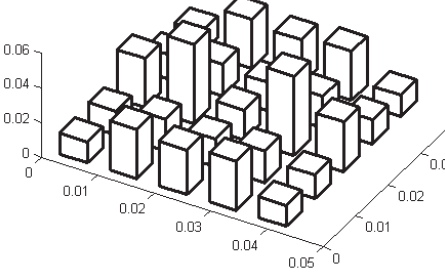
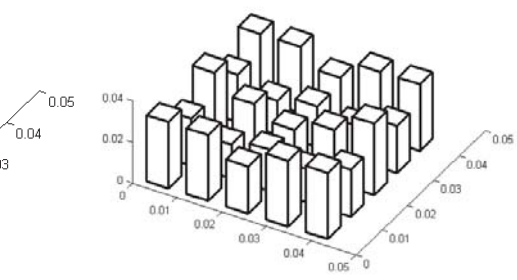
ตารางที่ 9-1 ผลการเปรียบเทียบ HV และ GD

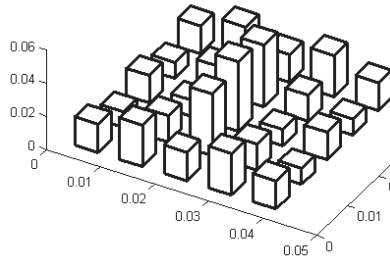
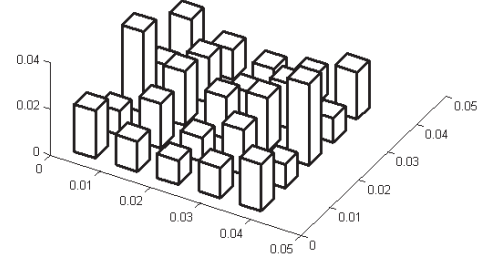
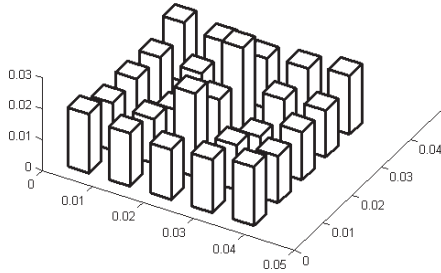
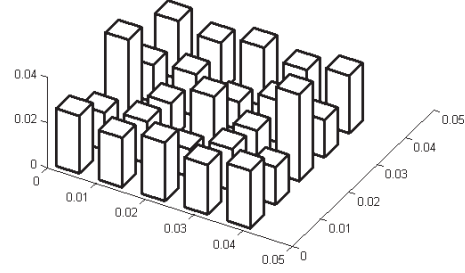
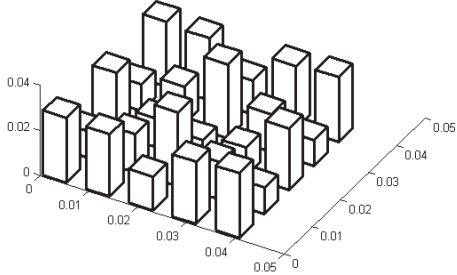
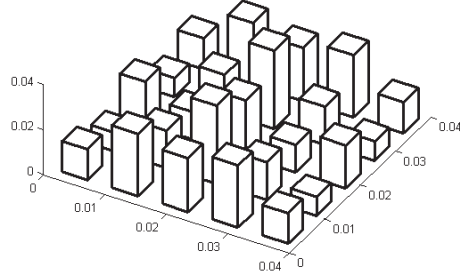
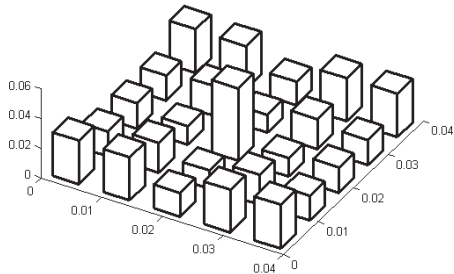
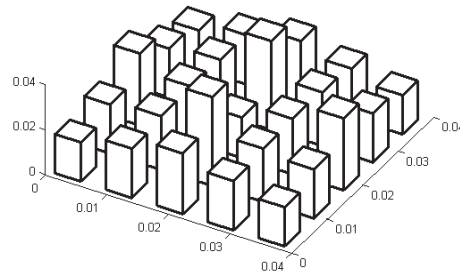
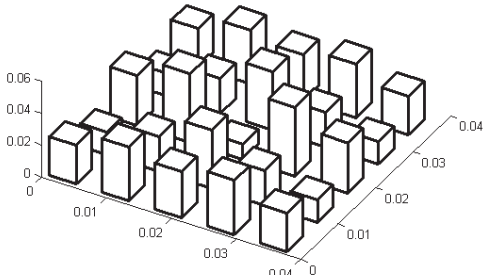
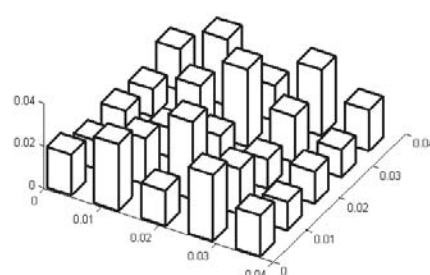
MOEAs	Average HV	Average GD
SPEA	0.8024	0.0438
PBIL	0.8141	0.0395
PSO	0.6427	0.0839
AMOSA	0.7788	0.0504

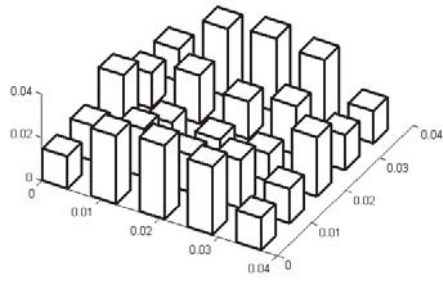
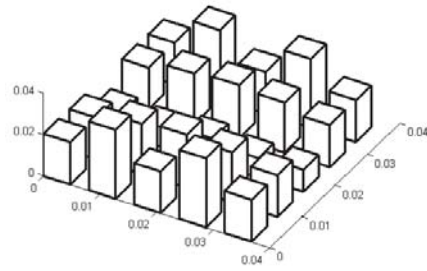
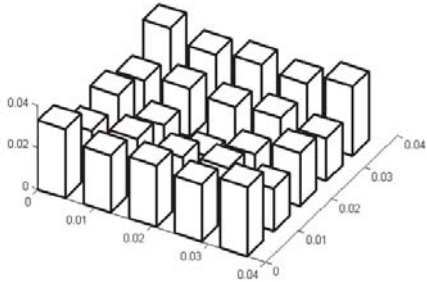
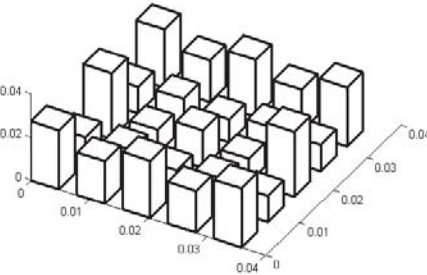
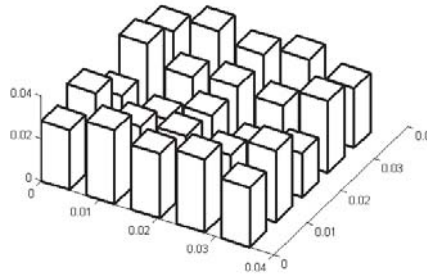
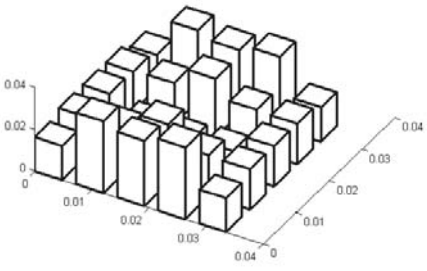
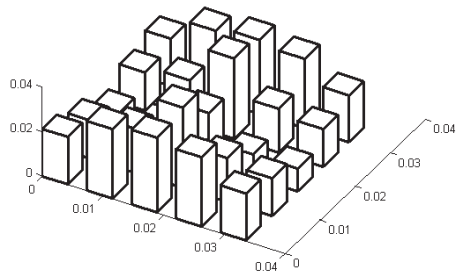
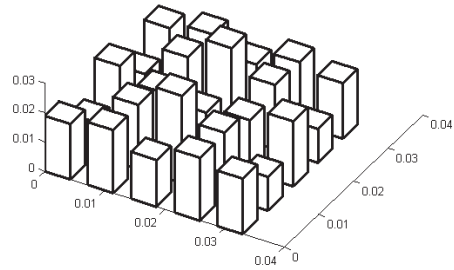
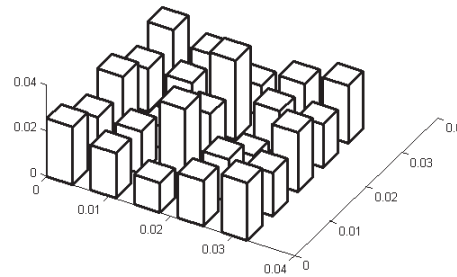
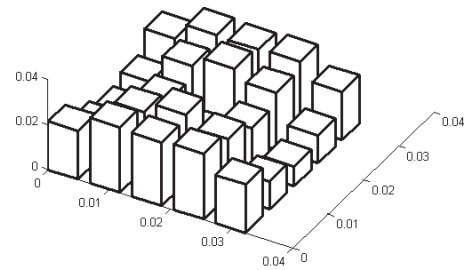
แนวพาเรโตที่ดีที่สุดซึ่งได้จากการรัน PBIL แสดงในรูปที่ 9-5 ส่วนขนาดและรูปทรงของอุปกรณ์ระบายความร้อนที่สอดคล้องกันแนวพาเรโตในรูป แสดงในรูปที่ 9-6 จากรูปแสดงให้เห็นว่า ในการรัน

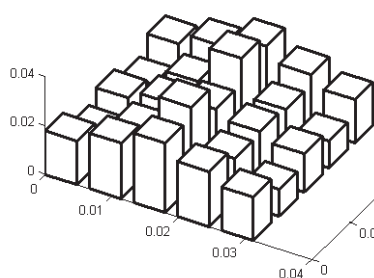
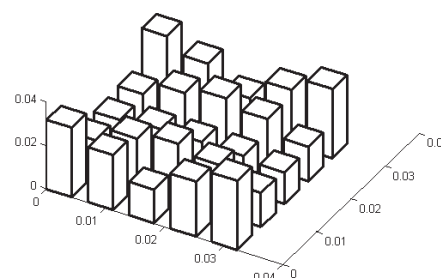
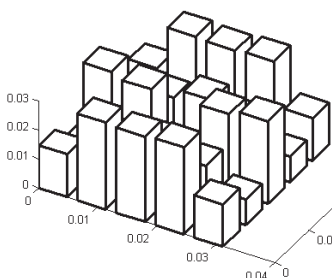
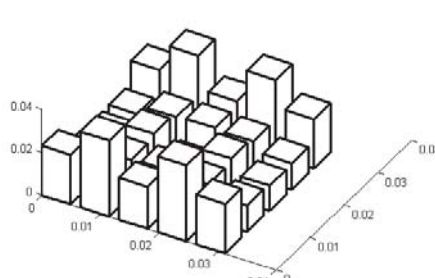
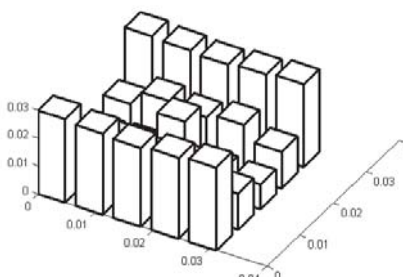
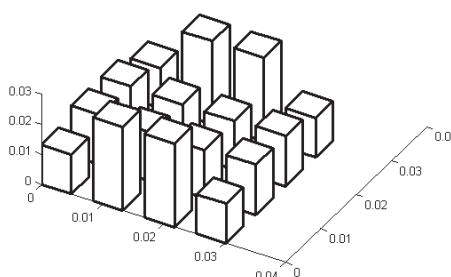
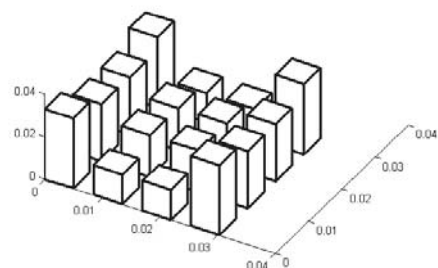
ขั้นตอนวิธีวิวัฒนาการหนึ่งครั้ง เราได้ผลเฉลยอุปกรณ์ระบายความร้อนแบบครีบท่ตรง หลากหลายรูปแบบดังแสดง



Solution#1 ($t_b=0.00455$)Solution#2 ($t_b=0.00487$)Solution#3 ($t_b=0.00458$)Solution#4 ($t_b=0.00465$)Solution#5 ($t_b=0.00485$)Solution#6 ($t_b=0.00486$)Solution#7 ($t_b=0.00487$)Solution#8 ($t_b=0.00486$)Solution#9 ($t_b=0.00486$)Solution#10 ($t_b=0.00488$)Solution#11 ($t_b=0.00492$)Solution#12 ($t_b=0.00457$)

Solution#13 ($t_b=0.00499$)Solution#14 ($t_b=0.00458$)Solution#15 ($t_b=0.00457$)Solution#16 ($t_b=0.00486$)Solution#17 ($t_b=0.00487$)Solution#18 ($t_b=0.00461$)Solution#19 ($t_b=0.00490$)Solution#20 ($t_b=0.00490$)Solution#21 ($t_b=0.00488$)Solution#22 ($t_b=0.00457$)

Solution#23 ($t_b=0.00488$)Solution#24 ($t_b=0.00486$)Solution#25 ($t_b=0.00488$)Solution#26 ($t_b=0.00486$)Solution#27 ($t_b=0.00454$)Solution#28 ($t_b=0.00492$)Solution#29 ($t_b=0.00488$)Solution#30 ($t_b=0.00490$)Solution#31 ($t_b=0.00488$)Solution#32 ($t_b=0.00491$)

Solution#33 ($t_b=0.00492$)Solution#34 ($t_b=0.00457$)Solution#35 ($t_b=0.00490$)Solution#36 ($t_b=0.00486$)Solution#37 ($t_b=0.00492$)Solution#38 ($t_b=0.00493$)Solution#39 ($t_b=0.00486$)

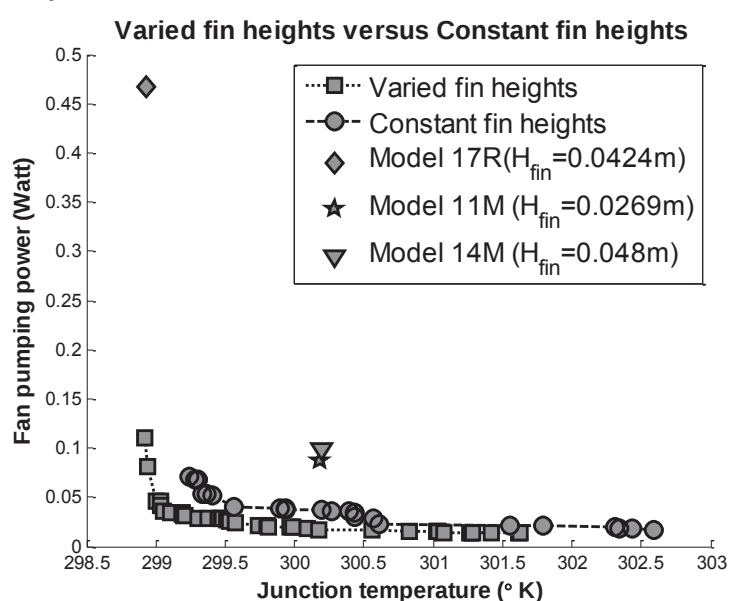
รูปที่ 9-6 รูปร่างของอุปกรณ์ระบายความร้อนที่เป็นผลเฉลยของพารโต

เพื่อตรวจสอบและยืนยันประสิทธิภาพของกระบวนการออกแบบที่คิดค้นขึ้นมาใหม่นี้ ได้ทำการหาค่าเหมาะที่สุดของอุปกรณ์ระบายความร้อนที่เป็นปัญหาการออกแบบเดียวกัน แต่ใช้ความสูงของครีบกังที่ 5 cm นอกจากนี้ ยังทำการวิเคราะห์หาค่าฟังก์ชันเป้าหมายของอุปกรณ์แลกเปลี่ยนความร้อนแบบครีบกังตรงที่มีใช้จริง ดังนี้

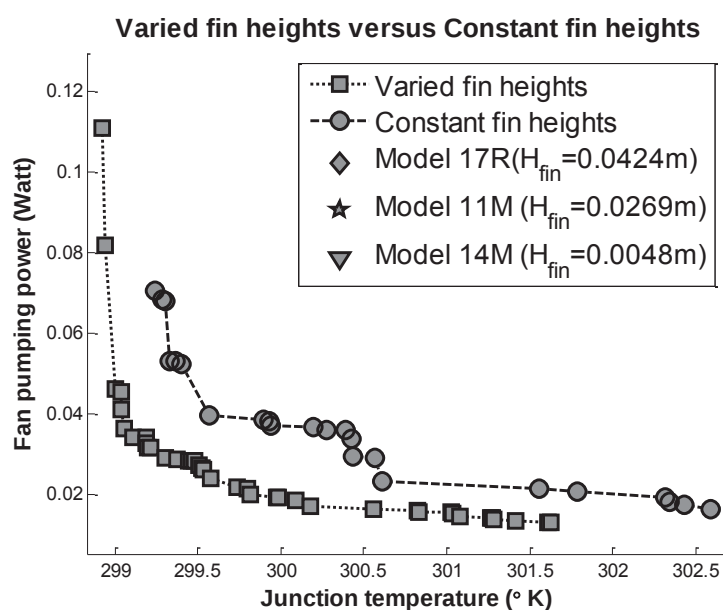
- **Model 17R** มีความกว้างฐานเป็น 0.0635m ความสูงของครีบกังที่ 0.0424m ขนาดของครีบกัง 0.0033m ความหนาของแผ่นฐาน 0.0127m และมีจำนวนครีบกังเป็น 14×14 fins
- **Model 11M** มีความกว้างฐานเป็น 0.0508m ความสูงของครีบกังที่ 0.0269m ขนาดของครีบกัง 0.0033m ความหนาของแผ่นฐาน 0.0048m และมีจำนวนครีบกังเป็น 61

- **Model 14M** มีความกว้างฐานเป็น 0.0508m ความสูงของครีบกึ่งที่ 0.0480m ขนาดของครีบกึ่ง 0.0033m ความหนาของแผ่นฐาน 0.0048m และมีจำนวนครีบกึ่งเป็น 61

เมื่อนำแนวพาเรโตที่ได้จากการออกแบบที่เหมาะสมที่สุดที่มีความสูงของครีบกึ่งที่ มาเปรียบเทียบกับแนวพาเรโตจากรูปที่ 9-5 และค่าฟังก์ชันเป้าหมายของอุปกรณ์ที่มีใช้งานจริงดังที่กล่าวในข้างต้น ผลการเปรียบเทียบแสดงในรูปที่ 9-7 และการขยายภาพของแนวพาเรโตอยู่ในรูปที่ 9-8 จากรูปทั้งสอง สามารถสรุปได้ว่า การหาค่าที่เหมาะสมที่สุดของอุปกรณ์ระบายความร้อนแบบครีบกึ่งตรง ที่มีการปรับค่าความสูงให้หลากหลาย จะให้ผลการออกแบบที่มีสมรรถนะทางความร้อนและค่าการดำเนินการที่ดีกว่าการออกแบบที่ใช้ความสูงของครีบกึ่งที่ และการหาค่าที่เหมาะสมที่สุดนี้ ทำให้ได้อุปกรณ์แลกเปลี่ยนความร้อนที่ดีกว่าอุปกรณ์รูปแบบเดียวกันที่มีใช้อยู่ในปัจจุบัน



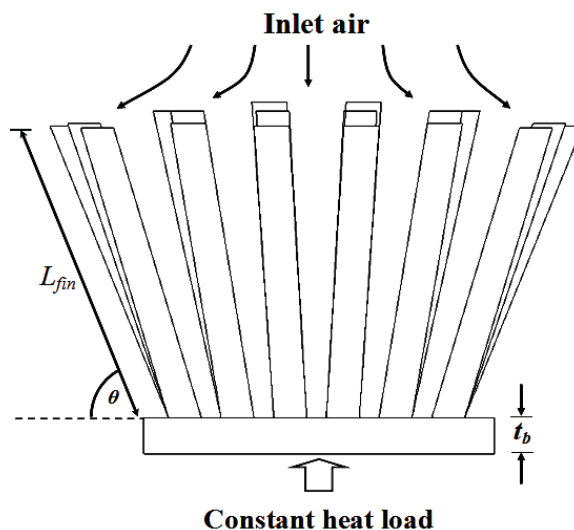
รูปที่ 9-7 เปรียบเทียบแนวพาเรโตจากการออกแบบที่มีการกระจายความสูงของครีบกึ่งและแบบความสูงคงที่



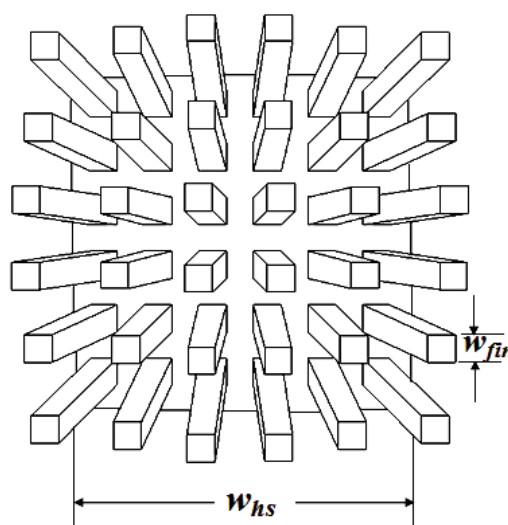
รูปที่ 9-8 ส่วนขยายของแนวพาเรโตในรูปที่ 9-7

9.3 การหาค่าเหมาะที่สุดแบบหลายเป้าหมายของอุปกรณ์ระบายความร้อนแบบครีบทั้งเอียง

รูปที่ 9-9 และ 9-10 แสดงแบบจำลองของอุปกรณ์ระบายความร้อนแบบครีบทั้งเอียง ตัวแปรที่เพิ่มขึ้นมานอกเหนือจากอุปกรณ์แบบครีบทั้งตรงคือ มุมเอียง (bevel) ของครีบทั้ง ดังแสดงในรูป



รูปที่ 9-9 รูปด้านข้างของอุปกรณ์แบบครีบทั้งเอียง



รูปที่ 9-10 รูปด้านบนของอุปกรณ์แบบครีบทั้งเอียง

9.3.1 การเข้ารหัสและถอดรหัสตัวแปรออกแบบ

กระบวนการสำคัญในการออกแบบรูปร่างของอุปกรณ์ระบายความร้อน ด้วยขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายคือ การเข้ารหัสและถอดรหัสตัวแปรออกแบบ การเข้ารหัสคือการกำหนดตัวแปรออกแบบว่าจะใช้ตัวแปรเลขฐานสอง เลขจำนวนเต็ม หรือเลขจำนวนจริง ส่วนการถอดรหัสคือการแปรผลจากตัวแปรออกแบบไปเป็นขนาดและรูปทรงของอุปกรณ์ เพื่อสร้างแบบจำลองสำหรับวิเคราะห์เชิงตัวเลขต่อไป ในการศึกษาี้ ขนาดของเวกเตอร์ออกแบบคือ 24×1 หรือมีตัวแปรกำหนดขนาดและรูปทรงของอุปกรณ์ระบายความร้อนแบบครีบทั้งตรงทั้งสิ้น 24 ตัวแปร กระบวนการถอดรหัส สามารถทำได้ ดังนี้

HeatSinkGeometry

ตัวแปรขาเข้า $\mathbf{x}$ ขนาด 24×1

ข้อมูลขาออก ขนาดและรูปทรงของอุปกรณ์แลกเปลี่ยนความร้อน

1: กำหนดค่าพารามิเตอร์ต่างๆจากเอลิเมนต์ของ $\mathbf{x}$

$$w_{fin} = x_1, N_f = \text{round}(x_2), t_b = x_3, \mathbf{L} = \{L_1, \dots, L_{10}\} = \{x_4 - x_{13}\}, \{\beta_1, \dots, \beta_9\} = \{x_{14} - x_{22}\}, V_f = x_{23}, w_{hs} = x_{24}$$

$$2: \text{คำนวณ } s_{fin} = (w_{hs} - N_f \cdot w_{fin}) / (N_f - 1)$$

3: ถ้าเงื่อนไขคือ $s_{fin} > 0.004$ ดำเนินการ

$$3.1 \ s_{fin} > 0.004$$

$$3.2 \ N_f = \text{round}((w_{hs} + s_{fin}) / (w_{fin} + s_{fin}))$$

4: หรือถ้าเงื่อนไขคือ $s_{fin} < 0.0025$ หาจำนวนครีป N_f ใหม่

$$4.1 \ N_f = \text{round}((w_{hs} + 0.0025) / (w_{fin} + 0.0025))$$

$$4.2 \ s_{fin} = (w_{hs} - N_f \cdot w_{fin}) / (N_f - 1)$$

จบเงื่อนไขตรรกศาสตร์

5: สร้างจุดควบคุม 7×7 จุด เขียนแทนด้วย $\mathbf{x}_0, \mathbf{y}_0$ การกระจายความยาวของครีปที่จุดควบคุม $\mathbf{L}$, การกระจายค่ามุมอิมิทของครีปที่จุดควบคุม α , และการกระจายค่าของมุมเอียงของครีปที่จุดควบคุม β

6: ปรับเปลี่ยนค่าของมุมเอียง $\beta_i; i = 2, \dots, 9$

$$6.1 \ \beta_1 = \beta_1$$

$$6.2 \ \beta_2 = \beta_2, \beta_5 = \beta_5 \frac{\beta_2}{\beta_2 + \beta_5}$$

$$6.3 \ \beta_3 = \beta_3, \beta_6 = (\beta_6 + \beta_8) \frac{\beta_3}{\beta_3 + \beta_6 + \beta_8}, \beta_8 = \beta_8 \frac{\beta_3}{\beta_3 + \beta_6 + \beta_8}$$

$$6.4 \ \beta_4 = \beta_4, \beta_7 = (\beta_7 + \beta_9) \frac{\beta_4}{\beta_4 + \beta_7 + \beta_9}, \beta_9 = \beta_9 \frac{\beta_4}{\beta_4 + \beta_7 + \beta_9}$$

7: คำนวณค่าของ $\mathbf{H}_x, \mathbf{H}_y, \mathbf{H}_z$ ที่จุดควบคุม

$$7.1 \ H_{x,i} = L_i \cos \alpha_i \sin \beta_i$$

$$7.2 \ H_{y,i} = L_i \sin \alpha_i \sin \beta_i$$

$$7.3 \ H_{z,i} = L_i \cos \beta_i$$

8: ดำเนินการอินเตอร์โพลชันแบบใช้ฟังก์ชันฐานหลักแนวรัศมี

$$8.1 \ \mathbf{c}_x = \text{SurfSpline}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{H}_x)$$

$$8.2 \ \mathbf{c}_y = \text{SurfSpline}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{H}_y)$$

$$8.3 \ \mathbf{c}_z = \text{SurfSpline}(\mathbf{x}_0, \mathbf{y}_0, \mathbf{H}_z)$$

9: กำหนดพิกัดของครีปแท่ง $\mathbf{x}_1, \mathbf{y}_1$

10: คำนวณความยาวในแนวแกนของครีปแท่ง

$$10.1: \mathbf{H}_{1x} = \text{SplineEval}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{c}_x, \mathbf{x}_0, \mathbf{y}_0)$$

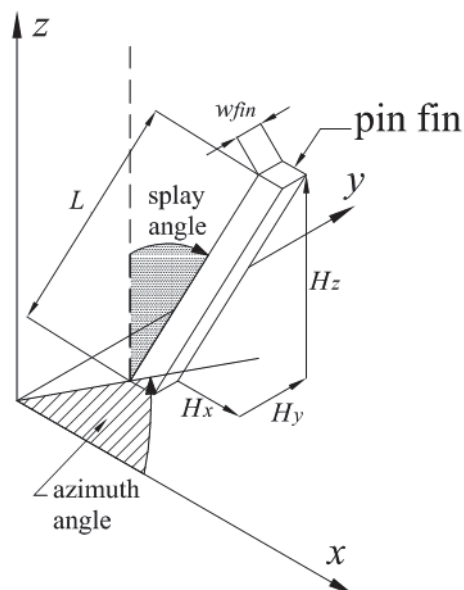
$$10.2: \mathbf{H}_{1y} = \text{SplineEval}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{c}_y, \mathbf{x}_0, \mathbf{y}_0)$$

$$10.3: \mathbf{H}_{1z} = \text{SplineEval}(\mathbf{x}_1, \mathbf{y}_1, \mathbf{c}_z, \mathbf{x}_0, \mathbf{y}_0)$$

11: สร้างรูปทรงของอุปกรณ์ระบายความร้อนแบบครีบแท่งเอียง จากข้อมูล $\mathbf{x}_1, \mathbf{y}_1, \mathbf{z}_1, \mathbf{H}_{1x}, \mathbf{H}_{1y}, \mathbf{H}_{1z}, N_f, w_{fin}, w_{hs}, t_b$.

12: บันทึกข้อมูลรูปทรงของอุปกรณ์เพื่อการวิเคราะห์ไฟในทัวลุ่ม

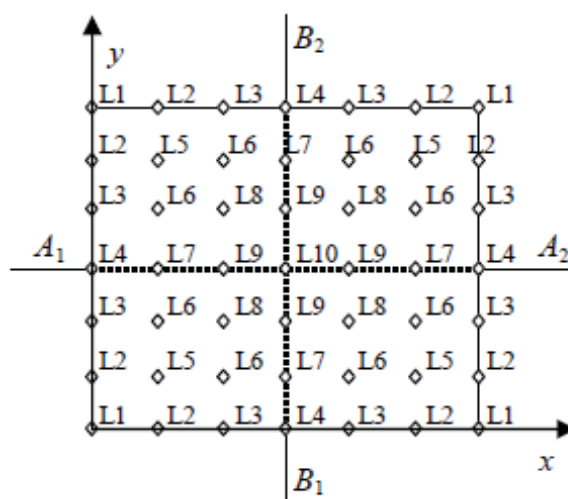
ในขั้นตอนที่ 1 ตัวแปร $x_1, \text{round}(x_2), x_3$ ถูกกำหนดให้เป็นความกว้างของอุปกรณ์ จำนวนครีบในแต่ละแถว และความหนาของฐานอุปกรณ์ตามลำดับ ตัวแปรที่ใช้ในการหาการกระจายความยาวของครีบที่จุดควบคุมคือ $L_i (x_4-x_{13})$ และสำหรับมุมเอียง $\beta_i (x_{14}-x_{22})$ รูปที่ 9-11 แสดงการสร้างครีบแท่งแบบเอียงและค่าพารามิเตอร์ที่ต้องใช้ในการระบุความเอียงของครีบ เราสามารถใช้เวกเตอร์กำหนดทิศทาง $H_x\bar{i} + H_y\bar{j} + H_z\bar{k}$ เพื่อระบุทิศทางเอียงของครีบใดๆ ส่วนประกอบของเวกเตอร์นี้สามารถหาได้จากความยาวของครีบ มุมอิมิต และมุมเอียง



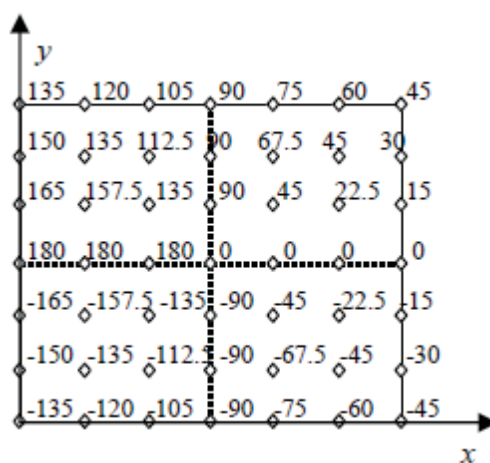
รูปที่ 9-11 พารามิเตอร์สำหรับสร้างอุปกรณ์ระบายความร้อนแบบครีบแท่งเอียง

ขั้นตอนที่ 2-4 เรียกว่าการซ่อมผลเฉลย กล่าวคือเมื่อสมาชิกบางตัวของผลเฉลยไม่อยู่ในขอบเขตที่กำหนดเอาไว้ เราสามารถประยุกต์ใช้กระบวนการซ่อมผลเฉลยในการบังคับให้ตัวแปรเหล่านั้นกลับเข้าไปอยู่ในขอบเขตที่ผู้ออกแบบกำหนดไว้ ซึ่งในที่นี้คือ $0.0025 \text{ m} \leq s_{fin} \leq 0.004 \text{ m}$.

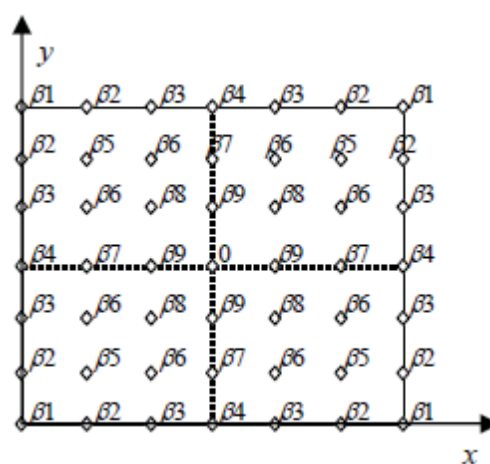
ขั้นตอนที่ 5 กำหนดตำแหน่งของจุดควบคุม ($\mathbf{x}_0, \mathbf{y}_0$) จำนวน $7 \times 7 = 49$ จุด ดังแสดงในรูปที่ 9-12 ซึ่งแสดงการกระจายค่าความยาวของครีบ (L) โดยมีเงื่อนไขว่า ความยาวครีบต้องสมมาตรเทียบกับระนาบ A_1-A_2 และ B_1-B_2 ในทำนองเดียวกัน การกระจายค่าของมุมอิมิต (α) และมุมเอียง (β) บนตำแหน่งของจุดควบคุมแสดงในรูปที่ 9-13 และ 9-14 ตามลำดับ



รูปที่ 9-12 การกระจายค่าความยาวของครีบริบที่จุดควบคุม



รูปที่ 9-13 การกระจายค่ามุมอิมิตของครีบริบที่จุดควบคุม



รูปที่ 9-14 การกระจายค่ามุมเอียงของครีบริบที่จุดควบคุม

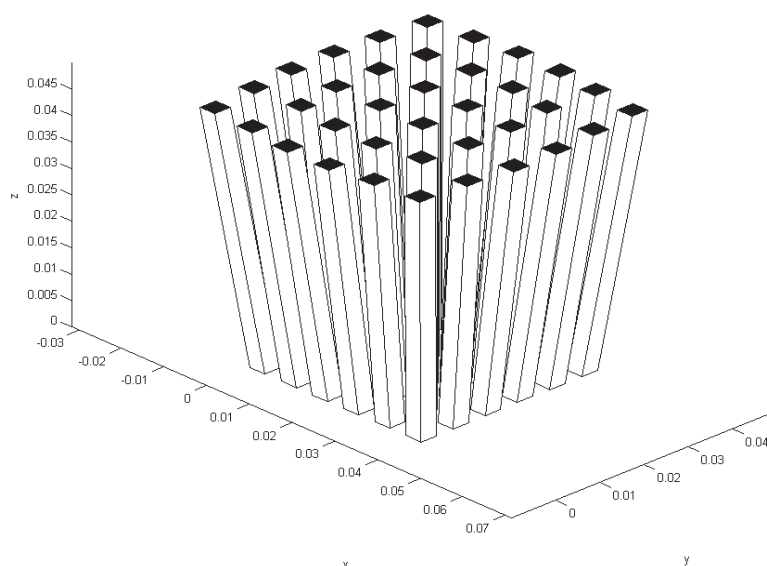
ขั้นตอนที่ 6 เป็นอีกขั้นตอนหนึ่งในการซ่อมผลเฉลย เพื่อให้ได้เงื่อนไข $\beta_5 \leq \beta_2$, $\beta_8 \leq \beta_6 \leq \beta_3$, และ $\beta_9 \leq \beta_7 \leq \beta_4$ ซึ่งเป็นเงื่อนไขที่ทำให้อุปกรณ์แลกเปลี่ยนความร้อนสามารถสร้างได้จริง

ขั้นตอนที่ 7 คำนวณการกระจายของส่วนประกอบย่อยของเวกเตอร์กำหนดทิศทางการเอียงของครีบริบที่จุดควบคุม (H_x , H_y , H_z)

ขั้นตอนที่ 8 หาสัมประสิทธิ์ของอินเตอร์โพลชันสำหรับ H_x , H_y , และ H_z (เขียนแทนด้วย c_x , c_y , และ c_z ตามลำดับ) โดยใช้การอินเตอร์โพลชันแบบฟังก์ชันฐานหลักแวนเดอร์สมิตดังแสดงในสมการ (9.2)

ขั้นตอนที่ 9 สร้างจุดที่เป็นตำแหน่งของกริปทั้งหมด x_1, y_1

ขั้นตอนที่ 10 หาส่วนประกอบย่อยของเวกเตอร์กำกับทิศทางของกริปแต่ละอันจากข้อมูลอินเตอร์โพลชันในขั้นตอนที่ 8 จากนั้น เราสามารถสร้างรูปร่างของอุปกรณ์ระบายความร้อนแบบกริปแท่งเอียงได้ในขั้นตอนต่อไป รูปที่ 9-15 แสดงตัวอย่างของอุปกรณ์ระบายความร้อนที่สร้างจากกระบวนการถอดรหัสตัวแปรออกแบบที่น่าเสนอ



รูปที่ 9-15 ตัวอย่างอุปกรณ์แบบกริปแท่งเอียงที่ได้จากกระบวนการถอดรหัสตัวแปรออกแบบ

9.3.2 ผลการรันขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมาย

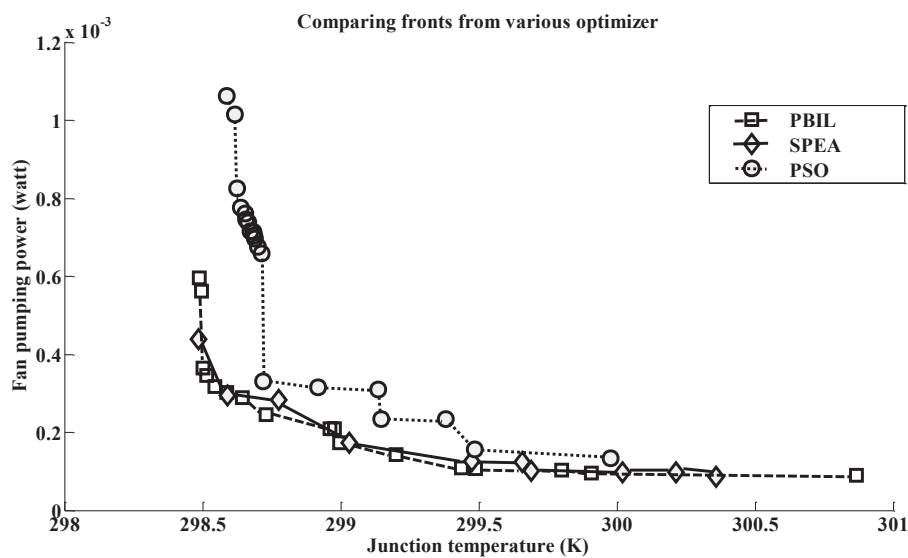
เมื่อได้กระบวนการถอดรหัสที่เหมาะสมแล้ว เราสามารถดำเนินการหาค่าฟังก์ชันในการออกแบบ โดยการวิเคราะห์ทางไฟไนท์วอลุ่ม และสามารถประยุกต์ใช้ขั้นตอนวิธีวิวัฒนาการที่ใช้เลขฐานสองหรือจำนวนจริงก็ได้ในการหาผลเฉลย ขั้นตอนวิธีวิวัฒนาการที่เลือกใช้เพื่อหาผลเฉลยของปัญหาการออกแบบที่เหมาะสมที่สุดในการศึกษานี้คือ PBIL, SPEA, และ PSO ทำการรันแต่ละขั้นตอนวิธีวิวัฒนาการ โดยใช้ขนาดประชากรเป็น 25 และจำนวนวงวนสูงสุดคือ 35 รอบ ซึ่งหมายความว่าแต่ละวิธีใช้จำนวนครั้งในการวิเคราะห์ไฟไนท์วอลุ่ม $35 \times 25 = 875$ ครั้ง ซึ่งใช้เวลาในการคำนวณมาก ขนาดของเมตริกซ์เก็บผลเฉลยที่ไม่ถูกรอรับคือ 50

ผลการรันแต่ละวิธีวิวัฒนาการแบบหลายเป้าหมายในการออกแบบอุปกรณ์นี้ แสดงในรูปที่ 9-16 การเปรียบเทียบค่า HV ของแนวพาเรโตในรูปที่ 9-16 แสดงในตารางที่ 9-2 จากผลการเปรียบเทียบพบว่า วิธีวิวัฒนาการแบบหลายเป้าหมายที่ดีที่สุด สำหรับปัญหาการออกแบบนี้ คือ PBIL รองลงมาคือ SPEA2 ส่วนวิธีที่ให้ผลการออกแบบเป็นอันดับสุดท้ายคือ PSO แนวพาเรโตที่ได้จากการรัน PBIL แสดงในรูปที่ 9-17

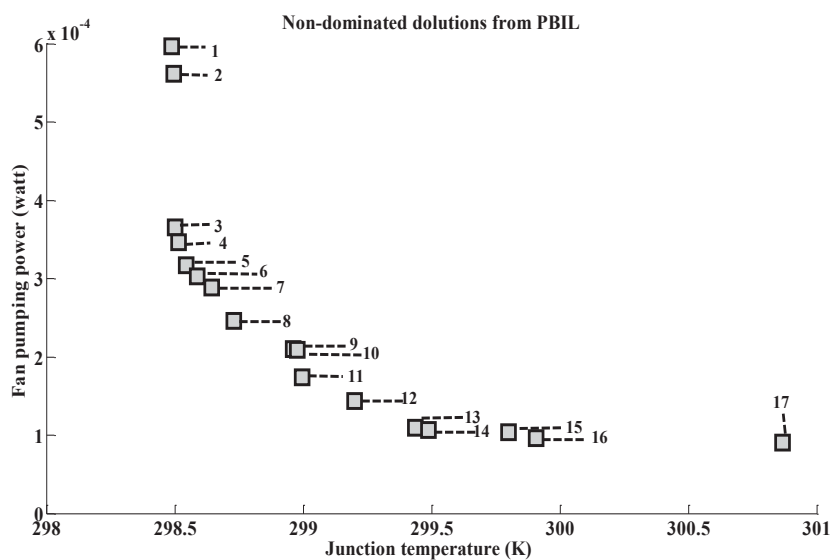
ส่วนรูปทรงของอุปกรณ์ระบายความร้อนที่เป็นผลเฉลยจากแนวพาเรโตนี้แสดงในรูปที่ 9-18 จากรูปพบว่า เราได้รูปทรงของอุปกรณ์หลากหลายรูปแบบ จากการรันขั้นตอนวิธีวิวัฒนาการหนึ่งครั้ง

ตารางที่ 9-2 ผลการเปรียบเทียบ HV

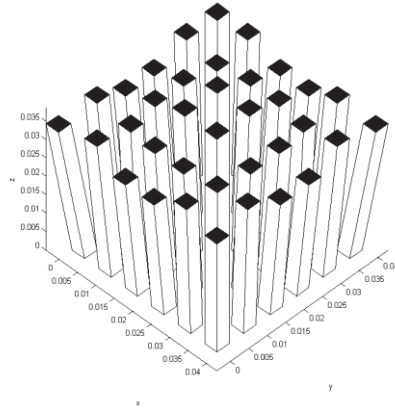
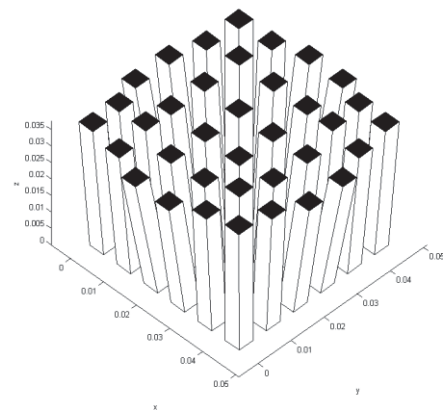
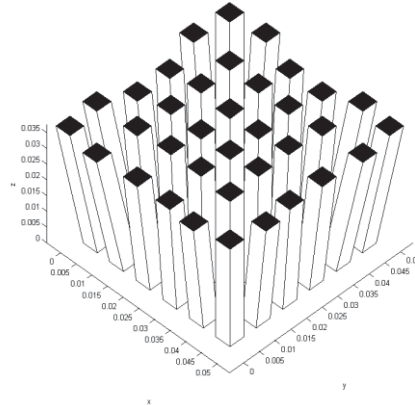
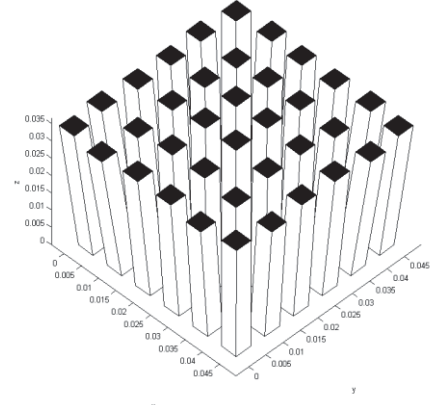
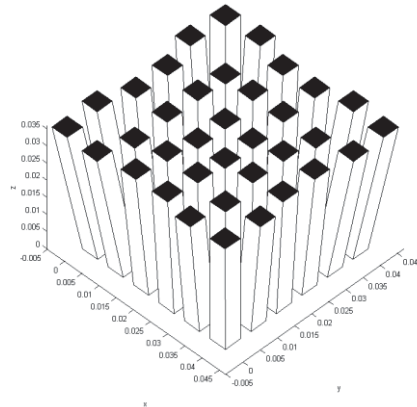
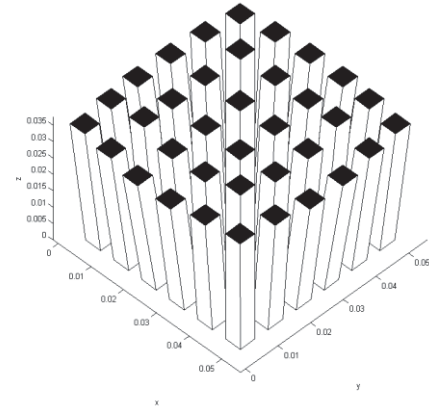
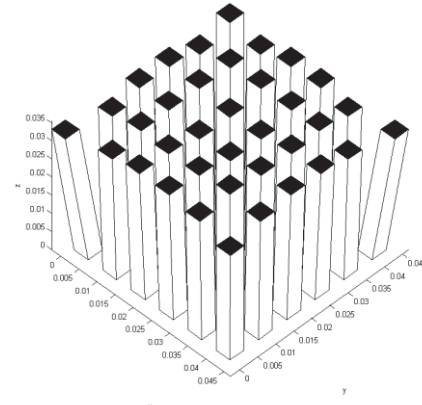
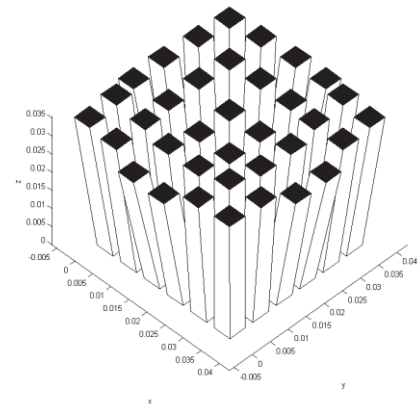
MOEAs	Hypervolume
SPEA	0.00228
PBIL	0.00231
PSO	0.00203

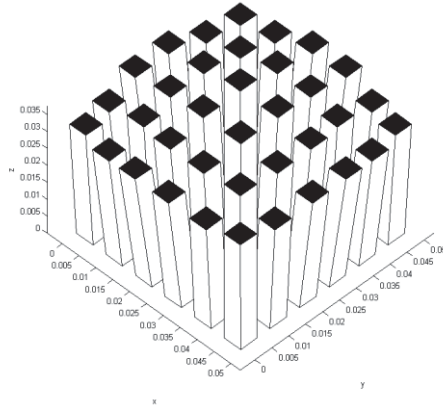
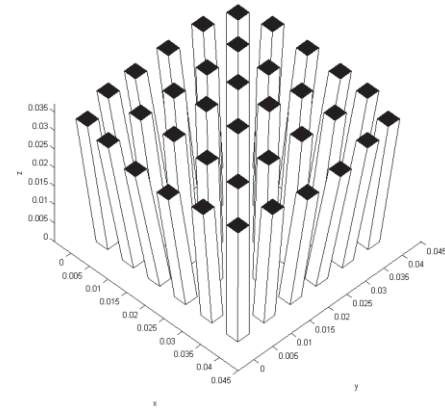
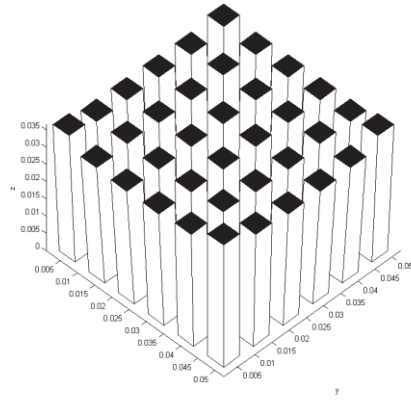
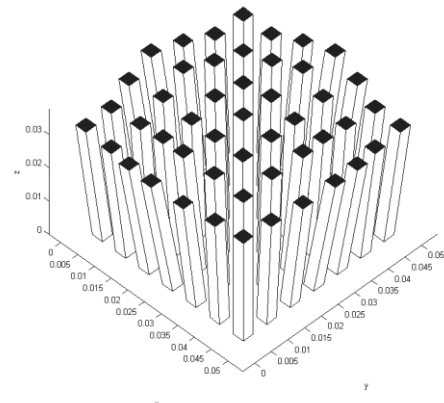
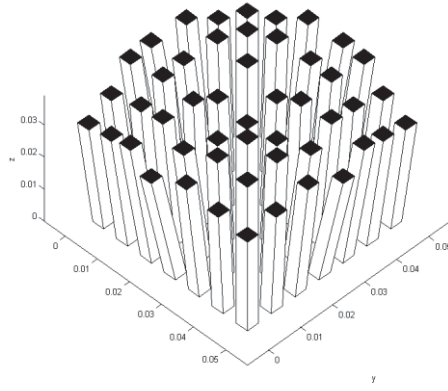
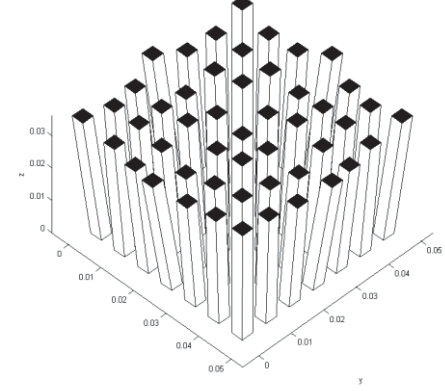
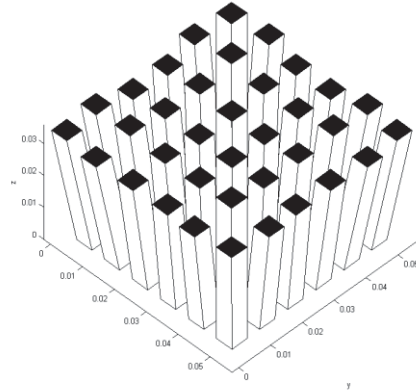
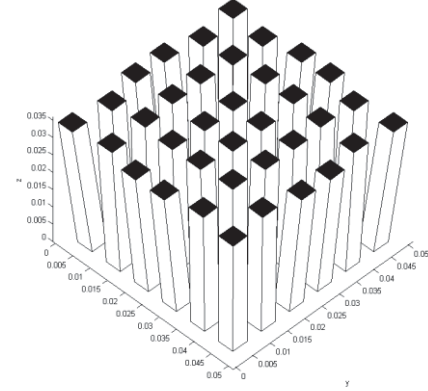


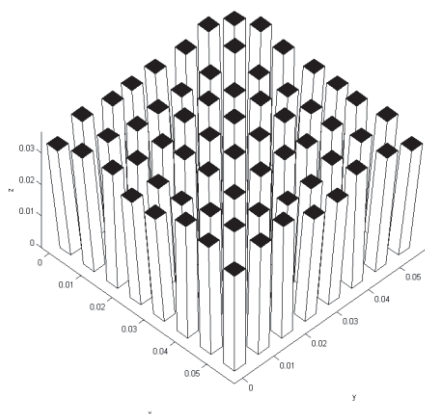
รูปที่ 9-16 ผลการรันด้วย PBIL, SPEA และ PSO



รูปที่ 9-17 แนวพาเรโตที่ได้จากการรัน PBIL

Solution#1 ($t_b=0.0049$)Solution#2 ($t_b=0.0037$)Solution#3 ($t_b=0.0049$)Solution#4 ($t_b=0.0049$)Solution#5 ($t_b=0.0042$)Solution#6 ($t_b=0.0039$)Solution#7 ($t_b=0.0031$)Solution#8 ($t_b=0.0027$)

Solution#9 ($t_b=0.0029$)Solution#10 ($t_b=0.0043$)Solution#11 ($t_b=0.0048$)Solution#12 ($t_b=0.0049$)Solution#13 ($t_b=0.0027$)Solution#14 ($t_b=0.0026$)Solution#15 ($t_b=0.0048$)Solution#16 ($t_b=0.0045$)

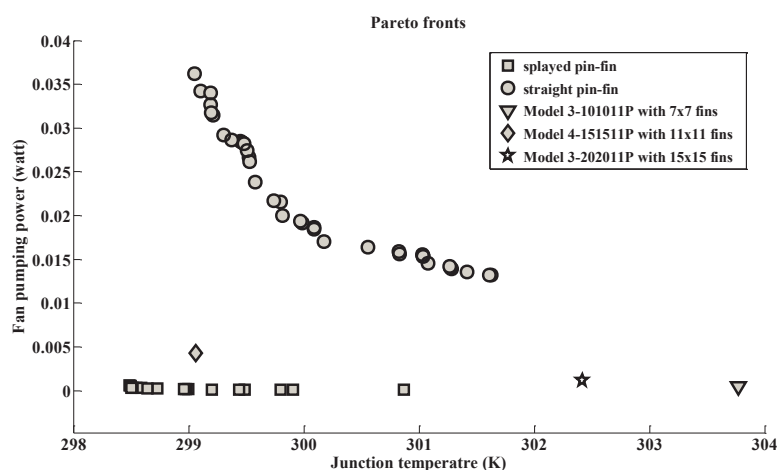
Solution#17 ($t_b=0.0033$)

รูปที่ 9-18 รูปทรงของอุปกรณ์ระบายความร้อนจากแนวพาเรโตในรูปที่ 9-17

เพื่อตรวจสอบและยืนยันประสิทธิภาพของกระบวนการออกแบบที่คิดค้นขึ้นมาใหม่นี้ ได้ทำการวิเคราะห์หาค่าฟังก์ชันเป้าหมายของอุปกรณ์แลกเปลี่ยนความร้อนแบบครีบท่ตรงที่มีใช้จริงในปัจจุบัน ดังนี้

- Model 3-101011P: 7×7 fins, 25.4 mm×25.4 mm fin base area, 27.9 mm fin length, 1.524 mm fin diameter, and 12° bevel angle
- Model 4-151511P: 11×11 fins, 38.1 mm×38.1 mm fin base area, 27.9 mm fin length, 1.778 mm fin diameter, and 12° bevel angle
- Model 3-202011P: 15×15 fins, 52.1 mm×52.1 mm fin base area, 27.9 mm fin length, 1.778 mm fin diameter, and 12° bevel angle

เมื่อนำแนวพาเรโตที่ได้จากการออกแบบที่เหมาะสมที่สุดในรูปที่ 9-5 มาเปรียบเทียบกับแนวพาเรโตจากรูปที่ 9-17 และค่าฟังก์ชันเป้าหมายของอุปกรณ์ที่มีใช้งานจริงดังที่กล่าวในข้างต้น ผลการเปรียบเทียบแสดงในรูปที่ 9-19 จากรูปพบว่า อุปกรณ์ที่ได้จากกระบวนการออกแบบในหัวข้อนี้ มีประสิทธิภาพสูงกว่าอุปกรณ์ที่มีใช้งานจริงในปัจจุบัน และดีกว่าการออกแบบในหัวข้อก่อนหน้านี้ หรืออาจกล่าวได้ว่า อุปกรณ์ที่ได้จากกระบวนการออกแบบในหัวข้อนี้มีสมรรถนะดีที่สุดเมื่อเทียบกับกระบวนการออกแบบในบทและหัวข้อที่ผ่านมา

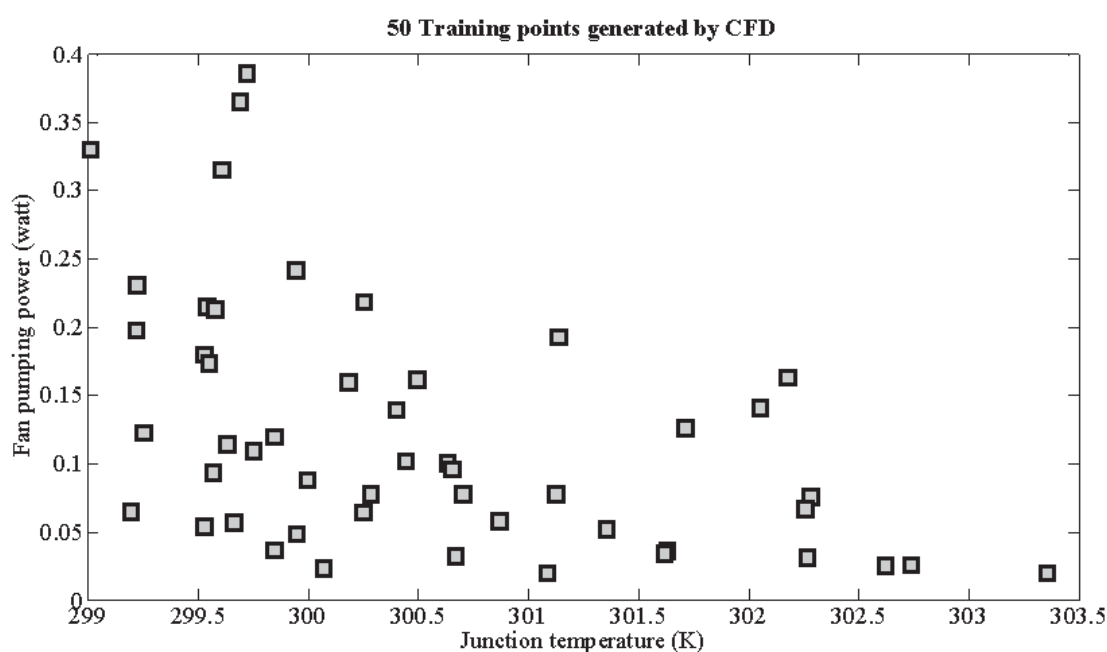


รูปที่ 9-19 การเปรียบเทียบผลการออกแบบกับอุปกรณ์ระบายความร้อนจริง

9.4 การประยุกต์ใช้ขั้นตอนวิธีพัฒนาการแบบผสมกับการออกแบบอุปกรณ์ระบายความร้อนแบบครีบท่ตรง

หัวข้อนี้ศึกษาการประยุกต์ใช้แบบจำลองทดแทนเพื่อเพิ่มสมรรถนะของการออกแบบที่เหมาะสมที่สุดของอุปกรณ์ระบายความร้อนแบบครีบท่ตรง ด้วยขั้นตอนวิธีพัฒนาการ ในที่นี้ จะเป็นการศึกษาการประยุกต์ใช้ขั้นตอนวิธีการผสมผสานแบบที่นำเสนอในหัวข้อย่อย 3.3.1 กล่าวคือใช้วิธี LHS สร้างเซตของผลเฉลยขึ้นมา จากนั้นสร้างแบบจำลองทดแทนในที่นี้คือแบบจำลอง Kriging หลังจากได้แบบจำลองทดแทน ทำการรันวิธี PBIL (เนื่องจากเป็นวิธีที่มีประสิทธิภาพสูงสุดในการออกแบบชนิดนี้) เมื่อได้ผลเฉลยของพารามิเตอร์แล้ว ทำการหาค่าฟังก์ชันจริงของผลเฉลยเหล่านั้น อันเป็นขั้นตอนสุดท้ายของกระบวนการออกแบบ

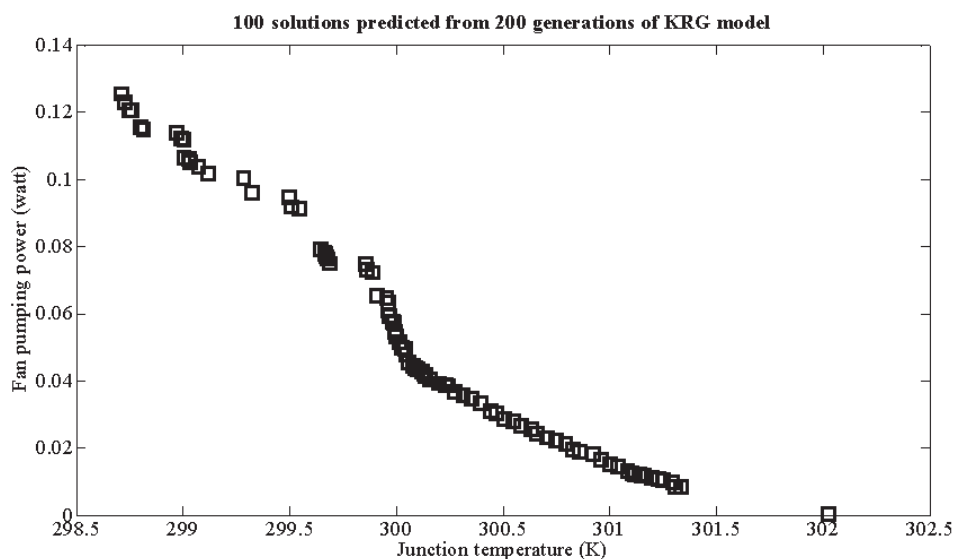
เงื่อนไขการออกแบบในการศึกษานี้คือ ประยุกต์ใช้ LHS สร้างผลเฉลย 50 ชุด เพื่อนำมาสร้างแบบจำลองทดแทน Kriging รูปที่ 9-20 แสดงตำแหน่งของผลเฉลยที่สุ่มสร้างขึ้นมาบนโดเมนของฟังก์ชันเป้าหมาย จากนั้นรัน PBIL ภายใต้การหาค่าฟังก์ชันด้วยแบบจำลองทดแทนโดยใช้ขนาดประชากร 100 และจำนวนวงวนเท่ากับ 200 เมื่อได้ผลเฉลยที่ไม่ถูกครอบงำแล้ว คัดเลือกผลเฉลยเหล่านั้นมา 25 ค่าเพื่อทำการหาค่าฟังก์ชันจริง และนำผลการออกแบบที่ได้ไปเปรียบเทียบกับผลเฉลยแนวพารามิเตอร์ในรูปที่ 9-5 ซึ่งถือว่าเป็นผลเฉลยที่ดีที่สุดสำหรับปัญหาการออกแบบนี้โดยการใช้ขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมาย



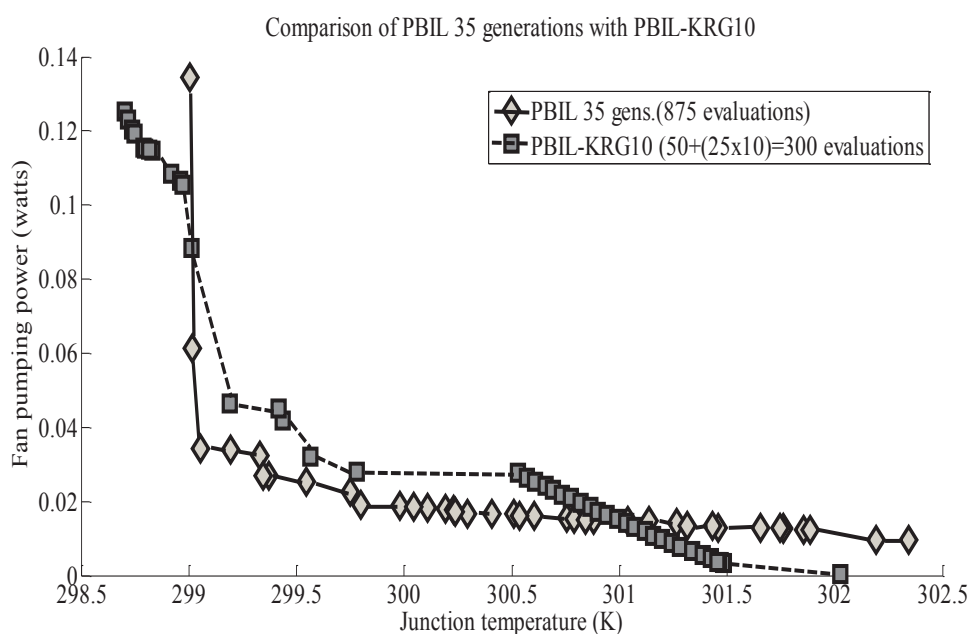
รูปที่ 9-20 ผลเฉลยสุ่มจาก LHS เพื่อสร้างแบบจำลองทดแทน

รูปที่ 9-21 แสดงแนวพารามิเตอร์ที่ได้จากการรัน PBIL โดยใช้การหาค่าฟังก์ชันเป้าหมายจากแบบจำลองทดแทน เมื่อทำการคัดผลเฉลยมา 25 ชุดและหาค่าฟังก์ชันเป้าหมายจริงของผลเฉลยเหล่านั้น จากการวิเคราะห์ทางไฟไนท์วอลุ่ม และนำไปเปรียบเทียบกับแนวพารามิเตอร์จากรูปที่ 9-5 ผลการเปรียบเทียบแสดงในรูปที่ 9-22 จากรูป แนวพารามิเตอร์ทั้งสองสลับกับครอบงำซึ่งกันและกันเป็นช่วงๆ แต่เมื่อคำนวณค่า HV พบว่า

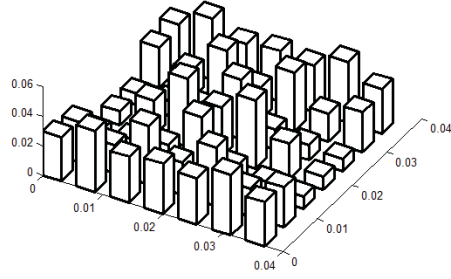
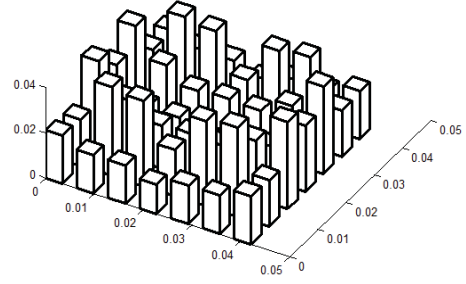
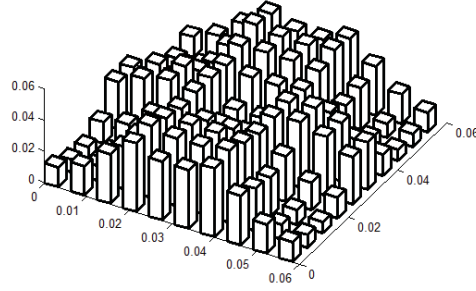
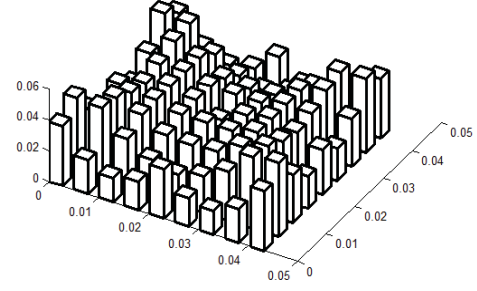
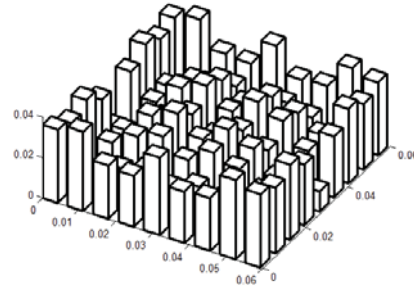
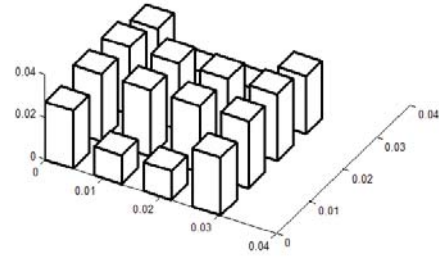
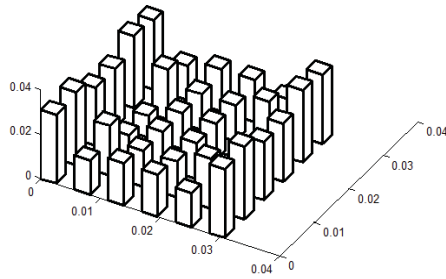
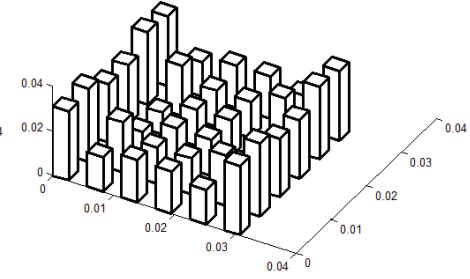
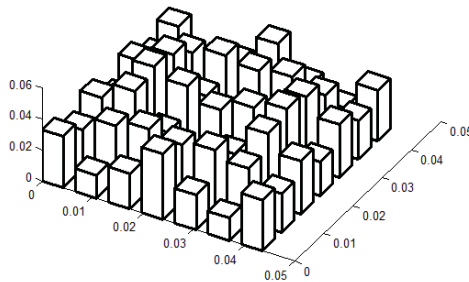
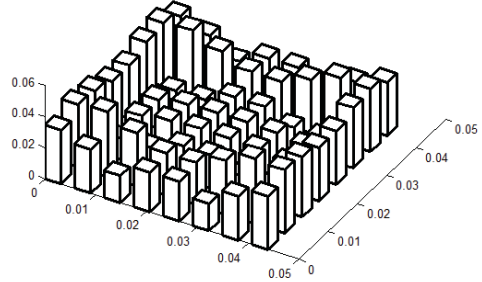
ค่า HV ของแนวพารโตรีปที่ 9-5 มีค่าเป็น 0.48699 ส่วน HV ของแนวพารโตรีที่ได้จากการศึกษาในหัวข้อนี้คือ 0.46845 แสดงว่าผลการออกแบบโดยใช้ PBIL ที่ไม่มีแบบจำลองทดแทนดีกว่าเล็กน้อย อย่างไรก็ตามเมื่อเปรียบเทียบจำนวนครั้งในการวิเคราะห์ไฟไนท์วอลุ่มพบว่า วิธี PBIL ที่ใช้แบบจำลองทดแทนวิเคราะห์ไฟไนท์วอลุ่มเพียง $50 + 25 = 75$ ครั้ง ในขณะที่แนวพารโตรีปที่ 9-5 ใช้ $35 \times 25 = 875$ ครั้ง นั้นหมายความว่าวิธีหลังใช้เวลาสั้นกว่าประมาณเกือบ 12 เท่า รูปทรงของอุปกรณ์ระบายความร้อนแบบครีปแท่งตรงที่ได้จากกระบวนการออกแบบในหัวข้อนี้แสดงในรูปที่ 9-23

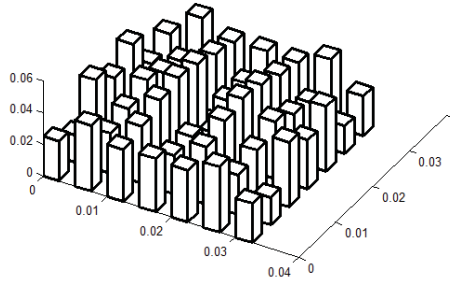
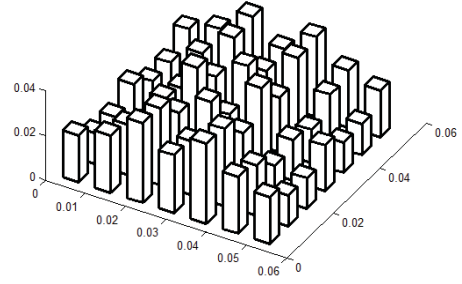
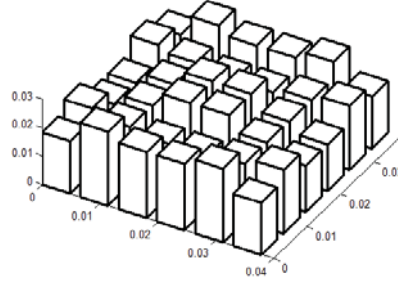
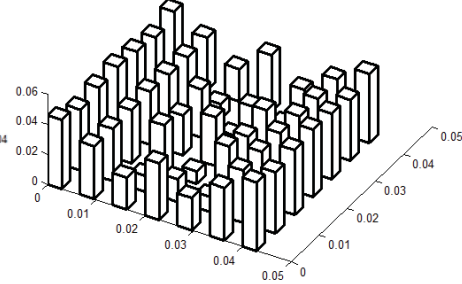
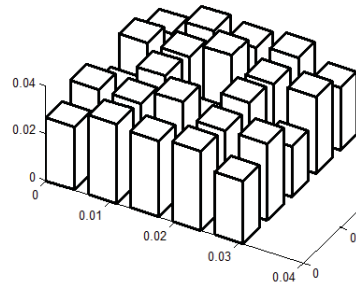
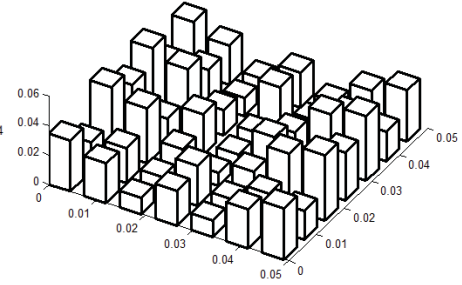
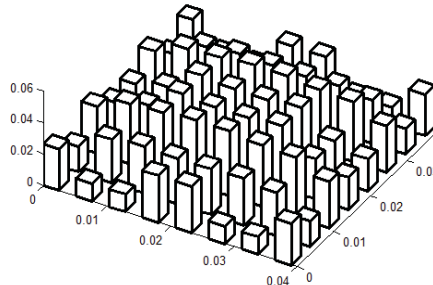
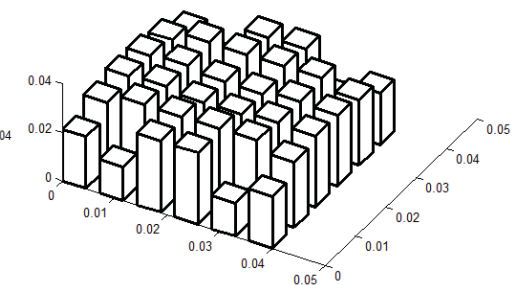
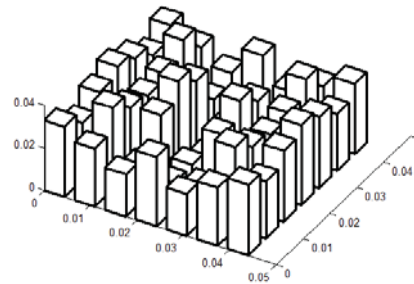
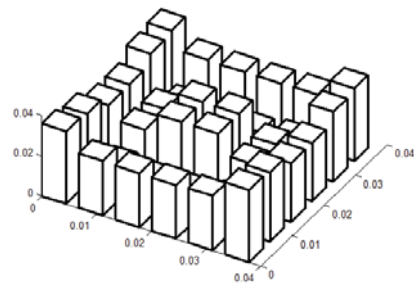


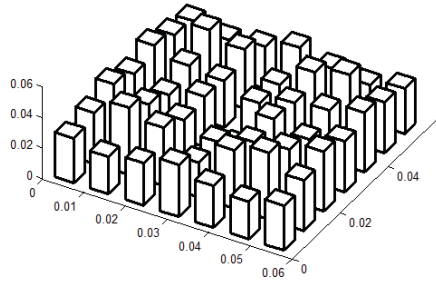
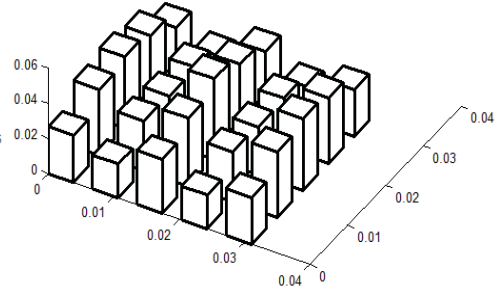
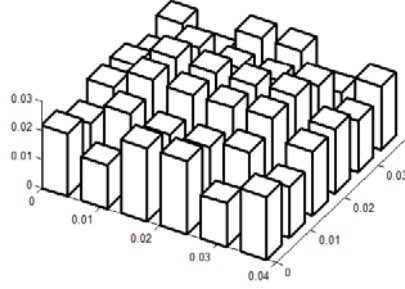
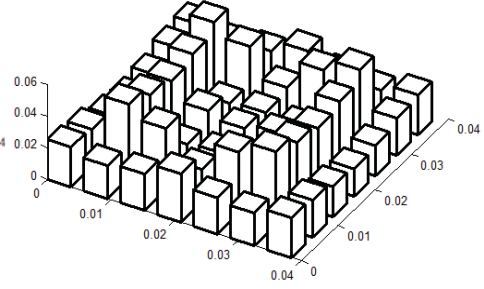
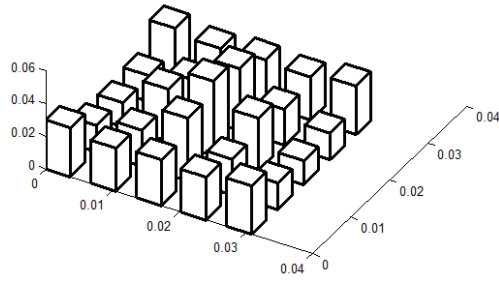
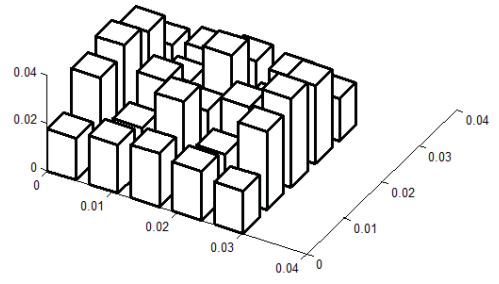
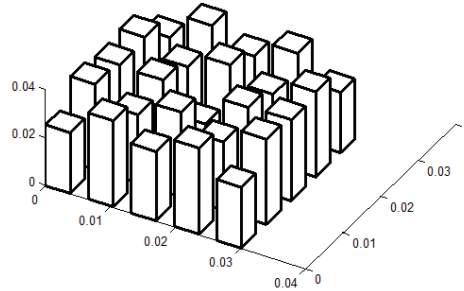
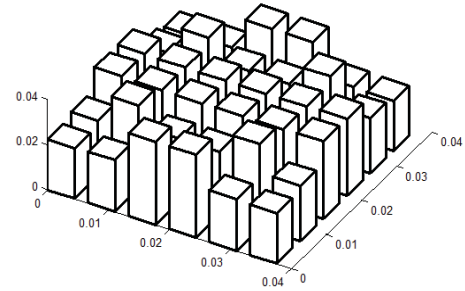
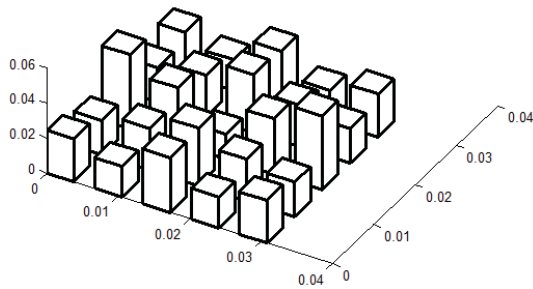
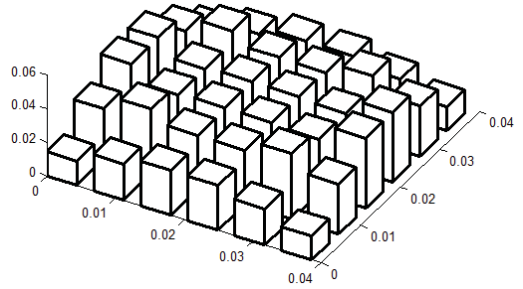
รูปที่ 9-21 แนวพารโตรีที่ได้จากการรัน PBIL โดยใช้แบบจำลองทดแทน Kriging

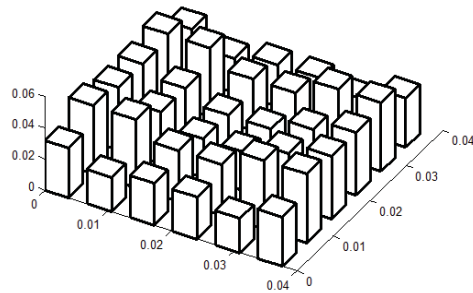
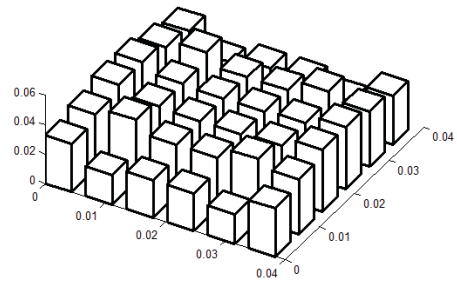
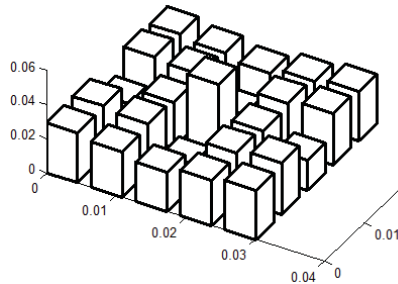
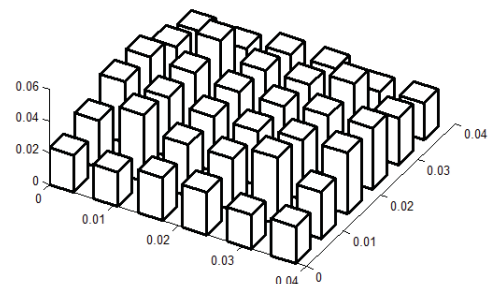
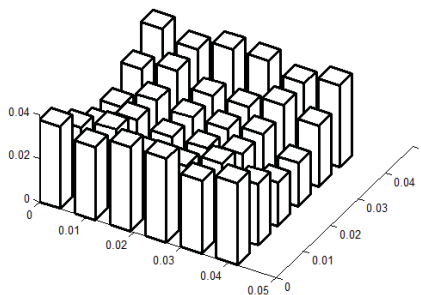
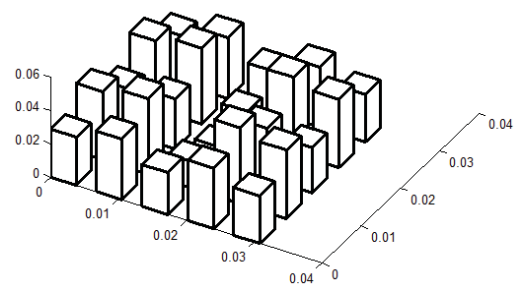
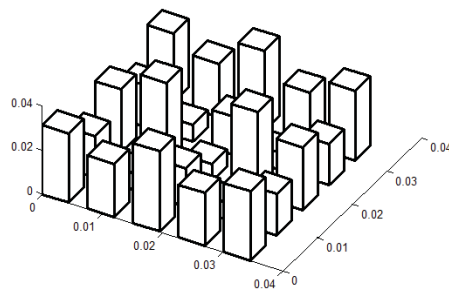
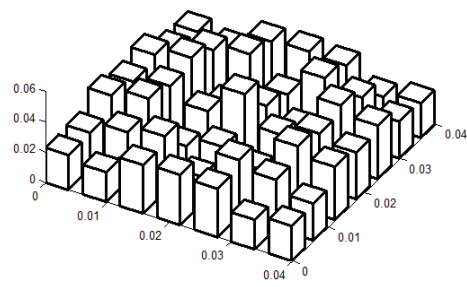
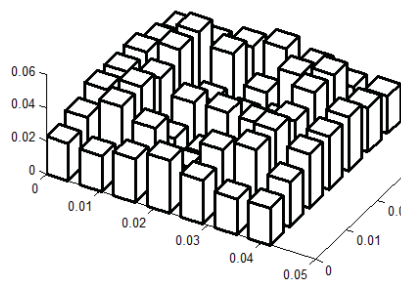
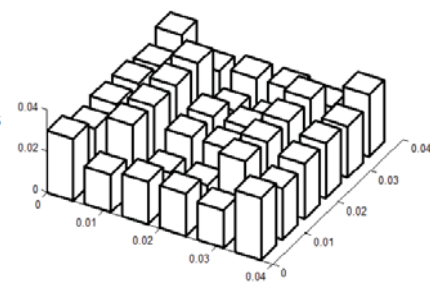


รูปที่ 9-22 การเปรียบเทียบแนวของผลเฉลย

Solution#1 ($t_b=0.00296$)Solution#2 ($t_b=0.00433$)Solution#3 ($t_b=0.00259$)Solution#4 ($t_b=0.00377$)Solution#5 ($t_b=0.00480$)Solution#6 ($t_b=0.00263$)Solution#7 ($t_b=0.00417$)Solution#8 ($t_b=0.00265$)Solution#9 ($t_b=0.00479$)Solution#10 ($t_b=0.00320$)

Solution#11 ($t_b=0.00266$)Solution#12 ($t_b=0.00386$)Solution#13 ($t_b=0.00386$)Solution#14 ($t_b=0.00258$)Solution#15 ($t_b=0.00256$)Solution#16 ($t_b=0.00260$)Solution#17 ($t_b=0.00258$)Solution#18 ($t_b=0.00385$)Solution#19 ($t_b=0.00385$)Solution#20 ($t_b=0.00308$)

Solution#21 ($t_b=0.00386$)Solution#22 ($t_b=0.00386$)Solution#23 ($t_b=0.00417$)Solution#24 ($t_b=0.00402$)Solution#25 ($t_b=0.00386$)Solution#26 ($t_b=0.00449$)Solution#27 ($t_b=0.00385$)Solution#28 ($t_b=0.00261$)Solution#29 ($t_b=0.00355$)Solution#30 ($t_b=0.00449$)

Solution#31 ($t_b=0.00479$)Solution#32 ($t_b=0.00423$)Solution#33 ($t_b=0.00386$)Solution#34 ($t_b=0.00401$)Solution#35 ($t_b=0.00446$)Solution#36 ($t_b=0.00479$)Solution#37 ($t_b=0.00401$)Solution#38 ($t_b=0.00402$)Solution#39 ($t_b=0.00433$)Solution#40 ($t_b=0.00433$)

รูปที่ 9-23 อุปกรณ์ระบายความร้อนที่ได้จากการหาผลเฉลยโดยใช้แบบจำลองทดแทน

สรุป วิจารณ์ และงานต่อเนื่อง

งานวิจัยนี้ศึกษาแนวทางการเพิ่มสมรรถนะในการหาผลเฉลี่ยของขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมาย สำหรับใช้ในการออกแบบ หาค่าเหมาะที่สุดแบบหลายเป้าหมายของระบบทางวิศวกรรมที่มีใช้งานจริง แนวทางเบื้องต้นของการวิจัยคือการนำเอาขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมายที่มีอยู่แล้ว และได้รับการพิสูจน์แล้วว่าสมรรถนะในการหาผลเฉลี่ยเหมาะที่สุดสูง เป็นวิธีหลัก จากนั้นทำการผสมผสานแบบจำลองทดแทนเข้าไปในกระบวนการคำนวณหลักของขั้นตอนวิธีเหล่านั้น

กระบวนการผสมผสานแบบจำลองทดแทนในขั้นตอนวิธีพัฒนาการสามารถทำได้หลายรูปแบบ แต่วิธีการหลักคือ ใช้ขั้นตอนวิธีพัฒนานั้นๆ เป็นกระบวนการคำนวณหลัก เมื่อได้ผลเฉลี่ยจำนวนมากพอที่จะสร้างแบบจำลองทดแทนจากการดำเนินการทางวิวัฒนาการ ทำการสร้างแบบจำลองทดแทนและใช้กระบวนการหาค่าเหมาะที่สุดด้วยแบบจำลองทดแทนนี้สร้างประชากรลูกกลุ่มพิเศษป้อนให้กับกระบวนการคำนวณของขั้นตอนวิธีพัฒนาการหลัก การดำเนินการในลักษณะนี้ทำให้สมรรถนะในการหาผลเฉลี่ยของขั้นตอนวิธีพัฒนาการแบบผสมมีสมรรถนะในการหาค่าตอบเพิ่มมากขึ้นอย่างมีนัยสำคัญ

ได้มีการศึกษาการนำผลเฉลี่ยประชากรพ่อ-แม่ และประชากรลูกมาคำนวณหาเวกเตอร์เกรเดียนต์โดยประมาณของตัวแปรออกแบบ และใช้เวกเตอร์เกรเดียนต์นั้นเป็นเวกเตอร์กำหนดทิศทางในการหาผลเฉลี่ย นอกจากนี้ ได้พัฒนาตัวดำเนินการที่ใช้ทิศทางวิวัฒนาการเพื่อใช้งานร่วมกับ เวกเตอร์เกรเดียนต์โดยประมาณ ผลการทดสอบสมรรถนะแสดงให้เห็นว่า เกรเดียนต์ประมาณที่คำนวณจากสมาชิกของประชากรในขั้นตอนวิธีพัฒนาการ มีประสิทธิภาพในการหาผลเฉลี่ยเหมาะที่สุดสูงมาก

ในส่วนของการประยุกต์ใช้งานขั้นตอนวิธีแบบผสมกับการออกแบบระบบจริงทางวิศวกรรม ได้ทำการประยุกต์ใช้ SPEA กับแบบจำลองทดแทนหลายรูปแบบเพื่อออกแบบชิ้นส่วนทางกล ในที่นี้คือแขนรับโมเมนต์บิดที่ทำจากโลหะแผ่น ผลจากการออกแบบและทดสอบสมรรถนะของแบบจำลองพบว่า แบบจำลองฟังก์ชันกำลังสองและแบบจำลอง Kriging ให้ผลเฉลี่ยพาราคติที่ดีที่สุด ทั้งนี้ เป็นเพราะว่าแบบจำลองทั้งสองทำนายค่าฟังก์ชันการออกแบบแม่นยำที่สุดในการออกแบบชิ้นส่วนยานยนต์และการออกแบบแขนรับโมเมนต์บิดสามมิติพบว่า การประยุกต์ใช้ PBIL ที่มีแบบจำลองทดแทนให้ผลการออกแบบดีกว่าการใช้ PBIL แบบไม่มีแบบจำลองทดแทนโดยที่ใช้จำนวนครั้งในการวิเคราะห์ทางไฟไนต์เอลิเมนต์น้อยกว่าอย่างมาก

การออกแบบรูปร่างของอุปกรณ์ระบายความร้อนเป็นกระบวนการออกแบบที่ทำได้ยาก ส่วนมากจะได้คำตอบจากการกำหนดรูปทรงของอุปกรณ์มาก่อน แล้วทำการหาขนาดเหมาะที่สุดเท่านั้น ในงานนี้ได้นำเสนอกระบวนการถอดรหัสตัวแปรออกแบบสำหรับอุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์ตรง อุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์เอียง อุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์ตรง และอุปกรณ์ระบายความร้อนแบบครีบบัฟเฟอร์เอียง

ร้อนแบบครีบท่งเอียง กระบวนการออกแบบอุปกรณ์ทั้งสี่นี้ จัดว่าเป็นทิศทางใหม่ในการออกแบบอุปกรณ์ระบายความร้อน ผลการออกแบบพบว่า อุปกรณ์ระบายความร้อนแบบที่สี่ มีสมรรถนะทางความร้อนและลดค่าดำเนินการได้มากที่สุด รองลงมาคืออุปกรณ์ระบายความร้อนแบบครีบท่งตรง และกระบวนการที่ได้อันดับที่สามคือ อุปกรณ์ระบายความร้อนแบบครีบท่งเอียง ผลการผสมผสานแบบจำลองทดแทนกับขั้นตอนวิธีวิวัฒนาการเพื่อการออกแบบอุปกรณ์นี้พบว่า การผสมผสานแบบจำลองทดแทนช่วยให้สมรรถนะของขั้นตอนวิธีวิวัฒนาการเหล่านั้นมีประสิทธิภาพสูงขึ้นและยังใช้จำนวนครั้งในการวิเคราะห์ไฟไนต์เอลูเมนต์น้อยกว่าการรันขั้นตอนวิธีวิวัฒนาการที่ไม่มีแบบจำลองทดแทนอย่างมาก

การขยายผลจากงานวิจัยนี้มีหลายทิศทาง แนวทางแรกคือการค้นคว้าวิจัยอย่างละเอียดเกี่ยวกับ การประยุกต์ใช้เวกเตอร์เกรเดียนต์โดยประมาณเพื่อเพิ่มสมรรถนะของขั้นตอนวิธีวิวัฒนาการทั้งแบบเป้าหมายเดียวและหลายเป้าหมาย ซึ่งปัจจุบันผู้วิจัยกำลังดำเนินการในส่วนนี้อยู่ นอกจากนี้ การผสมผสานวิธีซิมเพล็กซ์ลงเข้ากับขั้นตอนวิธีวิวัฒนาการก็เป็นประเด็นวิจัยอีกหัวข้อหนึ่งที่ควรศึกษา การประยุกต์ใช้การผสมขั้นตอนวิธีวิวัฒนาการแบบหลายเป้าหมายกับแบบจำลองทดแทนสำหรับการหาค่าเหมาะที่สุดที่มีเงื่อนไขบังคับปัจจุบันนี้ ยังเป็นกระบวนการออกแบบที่มีจุดด้อย เพราะเหตุว่า หากแบบจำลองทดแทนประมาณค่าของฟังก์ชันเงื่อนไขบังคับไม่แม่นยำพอ กระบวนการออกแบบขั้นตอนวิธีแบบผสมจะมีประสิทธิภาพลดลงอย่างมาก หรือแม้กระทั่งมีสมรรถนะต่ำกว่าการรันขั้นตอนวิธีวิวัฒนาการโดยไม่มีแบบจำลองทดแทน นอกจากนี้ การประยุกต์ใช้กระบวนการออกแบบผสมผสานนี้กับระบบจริงทางวิศวกรรม จะช่วยให้เกิดการพัฒนาต้นแบบหรือกระบวนการสร้างและขึ้นรูประบบเหล่านั้น ทั้งหมดนี้เป็นประโยชน์และสามารถประยุกต์ใช้ได้จริงในภาคอุตสาหกรรมของประเทศ

เอกสารอ้างอิง

- [1]. Abbass HA, Sarker R, Newton C. PDE: A Pareto–frontier Differential Evolution Approach for Multi-objective Optimization Problems. Proceedings of the 2001 Congress on Evolutionary Computation CEC2001
- [2]. Alonso JM, Alvarruiz F, Desantes JM, Hernández L, Hernández V, Moltó G. Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION 2007; 11: 46-55
- [3]. ARNOLD DV, BEYER HG. A Comparison of Evolution Strategies with Other Direct Search Methods in the Presence of Noise. Computational Optimization and Applications 2003; 24: 135–159
- [4]. Bäck T. Evolutionary Algorithms in Theory and Practice. Oxford University Press, Oxford, 1996
- [5]. Baluja S. Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Report CMU_CS_95_163, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1994
- [6]. Bandyopadhyay S, Saha S, Maulik U, Deb, K. A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION 2008; 12: 269-283
- [7]. Bar-Cohen A, Iyengar M. Least-energy optimisation of air-cooled heat sinks for sustainable development. IEEE Trans. Compon. Packag. Technol. 2003; 26: 16-25
- [8]. Basri M, Rahman RNZRA, Ebrahimpour A, Salleh AB, Gunawan ER, Rahman MBA. Comparison of estimation capabilities of response surface methodology (RSM) with artificial neural network (ANN) in lipase-catalyzed synthesis of palm-based wax ester. Biotechnology 2007; 7 (53)
- [9]. Bathe KJ. Finite Element Procedures. Prentice Hall, Englewood Cliffs, 1996
- [10]. Boonapan A, Bureerat S. Multi-stage design of an automotive component. International Journal of Vehicle Design 2011; in-press
- [11]. Büche D, Schraudolph NN, Koumoutsakos P. Accelerating Evolutionary Algorithms With Gaussian Process Fitness Function Models. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS 2005; 35: 183-194
- [12]. Buchholz P, Thümmler A. ENHANCING EVOLUTIONARY ALGORITHMS WITH STATISTICAL SELECTION PROCEDURES FOR SIMULATION OPTIMIZATION. Proceedings of the 2005 Winter Simulation Conference, 2005, pp.842-852
- [13]. Bureerat S. Introduction to Optimisation, Lecture note, Khon Kaen University, 2008
- [14]. Bureerat S. Hybrid population-based incremental learning using real codes. Lecture Notes in Computer Science 2011; 6683: 379-391.
- [15]. Bureerat S. Improved population-based incremental learning in continuous spaces. Advances in Intelligent and Soft Computing 2011; 96/2011: 77-86.
- [16]. Bureerat S, Cooper JE. Evolutionary algorithm using evolutionary direction and pseudo-gradient. 1st ASMO-UK, Ilkley, UK, 1998
- [17]. Bureerat S, Sriworamas K. Population-based incremental learning for multiobjective optimization. Advances in Soft Computing 2006; 39: 223-232
- [18]. Bureerat S, Srisoporn S. Optimum plate-fin heat sinks by using a multiobjective evolutionary algorithm. Engineering Optimization 2010; 42: 305-323.
- [19]. Cai Z, Wang Y. A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION 2006; 10: 658-675

- [20]. Chen MJ, Chen KN, Lin CH. Optimization on response surface models for the optimal manufacturing conditions of dairy tofu. *Journal of Food Engineering* 2005; 68: 471–480
- [21]. Cheng FY, Li D. Fuzzy set theory with genetic algorithms in constrained structural optimization. *ASCE Proceeding of US-Japan Joint Seminar on Structural Optimization*, New York, USA, 1997, pp. 55-66
- [22]. Chiang KT, Chang FP. Application of response surface methodology in the parametric optimisation of a pin-fin type heat sink. *International Communications in Heat and Mass Transfer* 2006; 33: 836–845
- [23]. Chiang KT, Chang FP, Tsai TC. Optimum design parameters of Pin-Fin heat sink using the grey-fuzzy logic based on the orthogonal arrays. *International Communications in Heat and Mass Transfer* 2006; 33: 744–752
- [24]. Coello CC. A short tutorial on evolutionary multiobjective optimization. Dept. of Electrical Engineering, Av. Instituto Politecnico Nacional No. 2508, Col. San Pedro Zacatenco
- [25]. Coello CAC (2002) List of references on evolutionary multi-objective optimization (journal papers only). www.jeo.org/emo/EMOOjournals.html
- [26]. Craig KJ, de Kock DJ, Gauchb P. Minimization of heat sink mass using CFD and mathematical optimization. *Journal of Electronic Packaging* 1999; 121: 143-147
- [27]. Culhamand JR, Muzychka YS. Optimization of plate fin heat sink using entropy generation minimization. *IEEETrans.Compon. Packag. Technol.* 2001; 24: 159-165
- [28]. Deb K, Pratap A, Agarwal S, Meyarivan T. A Fast and elitist multiobjective genetic algorithm: NSGAI. *IEEE Trans. On Evolutionary Computation* 2002; 6(2): 182-197
- [29]. Deb K, Pratap A, Meyarivan T. Constrained test problems for multi-objective evolutionary optimization. *Lecture Notes in Computer Science* 2001; 1993/2001: 284-298
- [30]. Emmerich MTM, Giannakoglou KC, Naujoks B. Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2006; 10: 421-439
- [31]. Fawaz Z, Xu YG, Behdinan K. Hybrid evolutionary algorithm and application to structural optimization. *Struct. Multidisc. Optim.* 2005; 30: 219 – 226
- [32]. Fleming PJ, Purshouse RC, Lygoe RJ. Many-Objective Optimization: An Engineering Design Perspective. *EMO2005, LNCS* 2005; 3410: 14–32
- [33]. Fan S-KS, Liang YC, Zahara E. A genetic algorithm and a particle swarm optimizer hybridized with Nelder–Mead simplex search. *Computers & Industrial Engineering* 2006; 50: 401–425
- [34]. Farina M, Amato P. Linked interpolation-optimization strategies for multicriteria optimization problems. *Soft computing.* 2005; 9: 54-65
- [35]. Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In: *Proc. Of the 5th Inter. Conf. on GAs*, 1993, pp. 416-423
- [36]. Giannakoglou KC, Papadimitriou DI, Karpolis IC. Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Comput. Methods Appl. Mech. Engrg.* 2006; 195: 6312–6329
- [37]. Goel T, Vaidyanathan R, Haftka RT, Shyy W, Queipo NV, Tucke K. Response surface approximation of Pareto optimal front in multi-objective optimization. *Comput. Methods Appl. Mech. Engrg.* 2007; 196: 879–893
- [38]. Harder RL, Desmarais RN. Interpolation Using Surface Splines. *Journal of Aircraft* 1972; 9(2): 189-191
- [39]. Herrera, F., Lozano, M., Molona, D.: Continuous Scatter Search: An Analysis of the Integration of Some Combination Methods and Improvement Strategies. *European Journal of Operational Research*, 169 (2006) 450-476

- [40]. Hilbert R, Janiga G, Baron R, Thevenin D. Multi-objective shape optimization of a heat exchanger using parallel genetic algorithms. *International Journal of Heat and Mass Transfer* 2006; 49: 2567–2577
- [41]. Horn J, Nafpliotis N, Goldberg DE. A niched pareto genetic algorithm for multiobjective optimization. In: *The 1st IEEE Conf. on Evolutionary Computation*, 1994, pp. 82-87
- [42]. Huband S, Hingston P, Barone L, While L. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2006; 10: 477-506
- [43]. Hussein MI, Hamza K, Hulbert GM, Scott RA, Saitou K. Multiobjective evolutionary optimization of periodic layered materials for desired wave dispersion characteristics. *Struct Multidisc Optim* 2006; 31: 60–75
- [44]. Ivvan S, Pena V, Rionda SB, Aguirre AH. Multiobjective Shape Optimization Using Estimation Distribution Algorithms and Correlated Information. *Evolutionary Multi-Criterion Optimization. In: EMO2005*, 2005, pp. 664-676
- [45]. Iyengarand M, Bar-Cohen A. Least-energy optimization of force convection plate-fin heat sinks. *IEEE Trans. Compon. Packag. Technol.* 2003; 26: 62-70
- [46]. Jin Y. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 2005; 9: 3–12
- [47]. Jin Y, Branke J. Evolutionary Optimization in Uncertain Environments—A Survey. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2005; 9: 303-317
- [48]. Kanyakam S, Bureerat S. Multiobjective evolutionary optimization of a splayed pin-fin heat sink. *Engineering Applications of Computational Fluid Mechanics* 2011; accepted.
- [49]. Kanyakam S, Bureerat S. Optimal design of a pin-fin heat sink using a surrogate-assisted multiobjective evolutionary algorithm. *Advanced Materials Research* 2011; 308-310: 1122-1128.
- [50]. Khan WA, Culham JR, Yovanovich MM. Optimisation of pin-fin heat sinks using entropy generation minimization. *2004 Inter Society Conference on Thermal Phenomena*, pp. 259-267
- [51]. Knowles J. ParEGO: A Hybrid Algorithm with On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2005; 10: 50-66
- [52]. Knowles J, Corne D. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimization. *Congress on Evolutionary Computation*, 1999
- [53]. Knowles JD, Corne DW. Approximating the non-dominated front using the pareto archive evolution strategy. *Evolutionary Computation*. 2000; 8(2): 149-172
- [54]. Krajnik P, Kopac J, Sluga A. Design of grinding factors based on response surface methodology. *Journal of Materials Processing Technology* 2005; 162–163: 629–636
- [55]. Krueger WB, Bar-Cohen A. Optimal numerical design of forced convection heat sinks. *IEEE Trans. Compon. Packag. Technol.* 2004; 27: 417-425
- [56]. Ktema HO Erzurumlu T, Kurtaran H. Application of response surface methodology in the optimization of cutting conditions for surface roughness. *Journal of Materials Processing Technology* 2005; 170: 11–16
- [57]. Kwana C-T, Janb F-S, Leea WC. An optimum die fillet design for minimizing the difference in extruded fin lengths of straight-fin heat sink extrusion. *Journal of materials processing technology* 2008; 201; 145–149
- [58]. Lophaven SN, Neilson HB, Sondergaard J. DACE a MATLAB Kriging Toolbox. Technical report IMM-TR-2002-12, 2002, Technical University of Denmark
- [59]. Mandal D, Pal SK, Saha P. Modeling of electrical discharge machining process using back propagation neural network and multi-objective optimization using non-

- dominating sorting genetic algorithm-II. *Journal of Materials Processing Technology* 2007; 186: 154–162
- [60]. Marler RT, Arora JS. Survey of multi-objective optimization methods for engineering. *Struct. Multidisc. Optim.* 2004; 26: 369 – 395
- [61]. Mayer DG, Kinghorn BP, Archer AA. Differential evolution – an easy and efficient evolutionary algorithm for model optimization. *Agricultural Systems* 2005; 83: 315–328
- [62]. McDonald DB, Grantham WG, Tabor WL, Murphy MJ. Global and local optimization using radial basis function response surface models. *Appl. Math. Model.* 2007; 31: 2095–2110
- [63]. Ong YS, Nair PB, Lum KY. Max–Min Surrogate-Assisted Evolutionary Algorithm for Robust Design. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2006; 10: 230–244
- [64]. OzcelikT B, Erzurumlu T. Determination of effecting dimensional parameters on warpage of thin shell plastic parts using integrated response surface method and genetic algorithm. *International Communications in Heat and Mass Transfer* 2005; 32: 1085–1094
- [65]. H. S. Park and X. P. Dang (2010). Structural optimisation based on CAD-CAM integration and metamodeling techniques. *Computer-Aided Design*.
- [66]. Park K, Moon S. Optimal design of heat exchangers using the progressive quadratic response surface model. *International Journal of Heat and Mass Transfer* 2005; 48: 2126–2139
- [67]. Park K, Oh PK, Lim HJ. The application of the CFD and Kriging method to an optimisation of heat sink. *International Journal of Heat and Mass Transfer* 2006; 49: 3439–3447
- [68]. N. Pholdee and S. Bureerat (2010). Passive vibration control of an automotive component using evolutionary optimisation. *The First TSME International Conference on Mechanical Engineering*, 20-22 October, 2010, Ubon Ratchathani, Thailand.
- [69]. Pholdee N, Bureerat S. Surrogate-assisted evolutionary optimizers for multiobjective design of a torque arm structure. *Applied Mechanics and Materials* 2011; in-press
- [70]. Purshouse RC, Fleming PJ. On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2007; 11: 770–784
- [71]. Regis RG, Shoemaker CA. Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2004; 8:490–505
- [72]. Reyes-Sierra, M., Coello Coello, C.A.: Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *Int. J. of Computational Intelligence Research*, 2:3 (2006) 287–308
- [73]. Ryu JH, Choi DH, Kim SJ. Numerical optimisation of the thermal performance of a micro-channel heat sink. *International Journal of Heat and Mass Transfer* 2002; 45: 2823–2827
- [74]. Schaffer JD. Multiobjective optimization with vector evaluated genetic algorithms. In: *GAs and their Application: Proc. of 1st Inter Conf. on GAs*, 1985, pp. 93–100
- [75]. Shan S, Wang GG. An efficient Pareto set identification approach for multi-objective optimization on black-box functions. *University of Manitoba*, 2004
- [76]. Shih CJ, Liu GC. Optimal design methodology of plate-fin heat sinks for electronic cooling using entropy generation strategy. *IEEE Trans. Compon. Packag. Technol* 2004; 27: 551–559

- [77]. Smith KI, Everson RM, Fieldsend JE, Murphy C, Misra R. Dominance-Based Multiobjective Simulated Annealing. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 2008; 13: 323-342
- [78]. Socha, K., Dorigo, M.: Ant Colony Optimization for Continuous Domains. *European Journal of Operational Research*, 185 (2008) 1155-1173
- [79]. Srinivas N, Deb K. Multiobjective optimization using non-dominated genetic algorithms. *Evolutionary Computation*. 1994; 2(3):221-248
- [80]. Srisomporn S, Bureerat S. Geometrical Design of Plate-Fin Heat Sinks Using Hybridization of MOEA and RSM. *IEEE Transactions on Components and Packaging Technologies* 2008; 31: 351-360
- [81]. Slesongsom S, Bureerat S. New Conceptual Design of Aeroelastic Wing Structures by Using Multi-objective Optimization, *Engineering Optimisation*, in-press, 2012
- [82]. Verrana GO, Mendesb RPK, Dalla Valentinaa LVO. DOE applied to optimization of aluminum alloy die castings. *Journal of materials processing technology* 2008; 200: 120–125
- [83]. Visser JA, de Kock DJ, Conradie FD. Minimization of heat sink mass using mathematical optimization. *Proc.16th IEEE SEM THERM Symp.*, 2000, pp. 252-259
- [84]. Wang N, Tai K. Handling Objectives as Adaptive Constraints for Multiobjective Structural Optimization. *CEC2007*, pp. 3922-3929
- [85]. Yıldız AR, Öztürk N, Kaya N, Öztürk F. Hybrid multi-objective shape design optimization using Taguchi's method and genetic algorithm. *Struct Multidisc Optim* 2007; 34: 317–332
- [86]. Yoo J, Lee CY. Topology optimization of a swing arm type actuator using the response surface method. *Microsystem Technologies* 2007; 13(1): 21-31
- [87]. Zhou Z, Ong YS, Lim MH, Lee BS. Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Comput* 2007; 11: 957–971
- [88]. Zhou Z, Ong YS, Nairy PB, Keaney AJ, Lumz KY. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 2007; 37: 66-76
- [89]. Zhou Z, Ong YS, Nguyen MH, Lim D. A Study on Polynomial Regression and Gaussian Process Global Surrogate Model in Hierarchical Surrogate-Assisted Evolutionary Algorithm. *CEC2005*, pp. 2832- 2839
- [90]. Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation* 2000; 8(2):173-195
- [91]. Zitzler E, Laumanns M, Bleuler S. A tutorial on evolutionary multiobjective optimization. *Computer Engineering and Network Laboratory, ETH, Zurich*
- [92]. Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary Methods for Design, Optimization and Control, Barcelona, Spain, 2002*
- [93]. Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. On Evolutionary Computation*. 1999; 3(4): 257-271
- [94]. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VG. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE trans. on Evolutionary Computation*. 2003; 7(2): 117-132

Output จากโครงการวิจัย

บทความที่ตีพิมพ์ในวารสารที่มี Impact factor

1. Srisoporn S, **Bureerat S**. Geometrical design of plate-fin heat sinks using hybridization of MOEA and RSM. IEEE Transactions on Components and Packaging Technologies 2008; 31: 351-360.
2. **Bureerat S**, Srisoporn S. Optimum plate-fin heat sinks by using a multiobjective evolutionary algorithm. Engineering Optimization 2010; 42: 305-323.
3. Kanyakam S, **Bureerat S**. Multiobjective evolutionary optimization of a splayed pin-fin heat sink. Engineering Applications of Computational Fluid Mechanics 2011; 5 (4) , pp. 553-565.

บทความที่ตีพิมพ์ในวารสารที่อยู่ในฐานข้อมูล SCOPUS แต่ไม่มี Impact factor

1. **Bureerat S**. Hybrid population-based incremental learning using real codes. Lecture Notes in Computer Science 2011; 6683: 379-391.
2. Pholdee N, **Bureerat S**. Surrogate-assisted evolutionary optimizers for multiobjective design of a torque arm structure. Applied Mechanics and Materials, 2011; 101-102 , pp. 324-328.
3. Kanyakam S, **Bureerat S**. Optimal design of a pin-fin heat sink using a surrogate-assisted multiobjective evolutionary algorithm. Advanced Materials Research 2011; 308-310: 1122-1128.

บทความอื่นๆ

1. **Bureerat S**. Improved population-based incremental learning in continuous spaces. Advances in Intelligent and Soft Computing 2011; 96/2011: 77-86.
2. Pholdee N, **Bureerat S**. Passive vibration control of an automotive component using evolutionary optimisation. TSME Journal of research and applications in mechanical engineering, 2011; 1(1), pp. 19-23.

บทความที่อยู่ระหว่าง จัดเตรียม และเสนอเพื่อพิจารณาตีพิมพ์ในวารสาร

1. Surrogate-assisted multiobjective evolutionary algorithms for structural shape and sizing optimisation
2. **Influence of Feasible and Infeasible Sampling Points on Constrained Structural Optimisation Using a Surrogate-Assisted Multiobjective PBIL**
3. Comparative performance of surrogate-assisted MOEAs for geometrical design of pin-fin heat sinks
4. Evolutionary approach for constructing an efficient surrogate model
5. A new sampling technique for surrogate-assisted constrained optimization using an approximate gradient

การนำผลงานวิจัยไปใช้ประโยชน์

- เจริญสาธารณะ ปัจจุบันยังไม่มีผลการวิจัยไปใช้ประโยชน์อย่างเป็นรูปธรรม แต่ในอนาคตจะมีการเผยแพร่รายงานผลการออกแบบชิ้นส่วนยานยนต์และอุปกรณ์ระบายความร้อนของ CPU คอมพิวเตอร์ ด้วยขั้นตอนวิธีพัฒนาการแบบหลายเป้าหมายไปสู่การใช้งานจริง
- เจริญวิชาการ คณาจารย์บัณฑิตด้านการออกแบบอุปกรณ์ระบายความร้อน 2 คน (จบการศึกษาแล้ว) และด้านการออกแบบชิ้นส่วนทางกล 1 คน (กำลังศึกษา)
- เจริญวิชาการ เนื้อหาและองค์ความรู้จากงานวิจัยนี้ ถูกบรรจุในวิชา Optimisation ซึ่งเป็นวิชาเรียนในระดับบัณฑิตศึกษา ภาควิชาวิศวกรรมเครื่องกล มหาวิทยาลัยขอนแก่น นอกจากนี้ยังนำองค์ความรู้และเนื้อหาเหล่านี้ บรรจุในหนังสือ “การหาค่าเหมาะที่สุดของระบบทางวิศวกรรมเครื่องกล” แต่งโดยผู้วิจัย ซึ่งจะทำให้การเผยแพร่ในปี 2555

ภาคผนวก

Geometrical Design of Plate-Fin Heat Sinks Using Hybridization of MOEA and RSM

Sungkom Srisomporn and Sujin Bureerat

Abstract—The work in this paper is aimed at demonstrating the practical multiobjective optimization of plate-fin heat sinks and the superiority of using a combined response surface method and multiobjective evolutionary optimizer over solely using the evolutionary optimizer. The design problem assigned is to minimize a heat sink junction temperature and fan pumping power. Design variables determine a heat sink geometry and inlet air velocity. Design constraints are given in such a way that the maximum and minimum fin heights are properly limited. Function evaluation is carried out by using finite volume analysis software. Two multiobjective evolutionary optimization strategies, real-code strength Pareto evolutionary algorithm with and without the use of a response surface technique, are implemented to explore the Pareto optimal front. The optimum results obtained from both design approaches are compared and discussed. It is illustrated that the multiobjective evolutionary technique is a powerful tool for the multiobjective design of electronic air-cooled heat sinks. With the same design conditions and an equal number of function evaluations, the multiobjective optimizer in association with the response surface technique totally outperforms the other. The design parameters affecting the diversity of the Pareto front include fin thickness, fin height distribution, and inlet air velocity while the plate base thickness and the total number of fins of the non-dominated solutions tend to approach certain values.

Index Terms—Finite volume method, multiobjective evolutionary algorithm, plate-fin heat sink, response surface method, strength Pareto evolutionary algorithm.

NOMENCLATURE

A	Set of non-dominated solutions or external Pareto archive.
A^k	External Pareto archive at the <i>k</i> th iteration for SPEA main procedure.
<i>b</i>	Fin-to-fin space.
B	Set of non-dominated solutions or external Pareto archive.
B^t	External Pareto archive at the <i>t</i> th iteration for SPEA using RSM model.
<i>C</i> (A , B)	Indicator comparing front A with respect to front B .

Manuscript received February 3, 2007; revised August 28, 2007. This work was supported by the Thailand Research Fund (TRF), Commission on Higher Education, and the Faculty of Engineering, Khon Kaen University. This work was recommended for publication by Associate Editor B. Courtois upon evaluation of the reviewers comments.

The authors are with the Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand (e-mail: sujbur@kku.ac.th).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAPT.2008.916799

<i>C</i> ₁	<i>C</i> -indicator of a Pareto front from SPEA-RSM with respect to a Pareto front from SPEA.
<i>C</i> ₂	<i>C</i> -indicator of a Pareto front from SPEA with respect to a Pareto front from SPEA-RSM.
<i>d</i>	Fin thickness.
f	Vector of objective functions.
<i>f</i> ₁	The first objective function (junction temperature).
<i>f</i> ₂	The second objective function (fan pumping power).
g	Vector of approximate objective functions.
<i>g</i> ₁	The approximate function of <i>f</i> ₁ .
<i>g</i> ₂	The approximate function of <i>f</i> ₂ .
<i>H</i>	Fin height.
<i>H</i> _{min} , <i>H</i> _{max}	Minimum and maximum fin heights.
<i>H</i> ₁ , ..., <i>H</i> ₄	Parameters indicating fin height distribution.
<i>k</i>	Iteration counter of SPEA at the main procedure.
K	Interpolation matrix.
<i>k</i> _{max}	Maximum iteration for SPEA main procedure.
K [#]	The pseudo-inverse of matrix K .
<i>L</i>	Plate base length.
<i>l_i</i>	Lower limit of <i>x_i</i> or <i>y_i</i> .
<i>m</i>	Number of objective functions.
<i>M</i>	Number of design variables (size of x and y).
<i>m</i> _a	Air mass flow rate.
<i>n</i>	Total number of fins.
<i>N</i>	Population size of SPEA.
<i>N_A</i>	Pareto archive size.
ΔP	Pressure drop across the heat sink.
<i>P_F</i>	Fan pumping power.
<i>p_i</i>	Interpolation coefficients of <i>g</i> ₁ .
<i>P_r</i>	Uniform random number.
P	Set of design solutions or population.
P ^{<i>k</i>}	Population at the <i>k</i> th iteration for SPEA main procedure.
<i>Q</i>	Heat flux.
<i>q_i</i>	Interpolation coefficients of <i>g</i> ₂ .
Q ^{<i>t</i>}	Population of design solutions at the <i>t</i> th iteration for SPEA using RSM.
rand ₁ , rand ₂	Uniform random number.

R_{HS}	Heat sink thermal resistance.
r_{ij}	Distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.
$RMSE_1$	Root mean square error of junction temperature.
$RMSE_2$	Root mean square error of fan pumping power.
t	Iteration counter of SPEA using RSM model.
t_b	Plate base thickness.
t_{max}	Maximum iteration for SPEA using RSM.
T_a	Ambient air temperature.
u_i	Upper limit of x_i or y_i .
V_f	Inlet air velocity.
W	Plate base width.
$\mathbf{x}, \mathbf{y}$	Vector of design variables.

Greek symbols

λ	Uniform random number.
ρ_a	Air density.
$\phi(r_{ij})$	Radial basis function.

ABBREVIATIONS

MOEAs	Multiobjective evolutionary algorithms.
NNT	Nearest neighborhood technique.
RMSE	Root mean square error.
RSM	Response surface method.
SPEA	Strength Pareto evolutionary algorithm.
SPEA-RSM	Strength Pareto evolutionary algorithm using a response surface method.

I. INTRODUCTION

DUE to the growth trend in integrated circuit (IC) technology recently, electronic cooling systems and components with higher performance are needed. Higher power density and heat dissipation as well as a decreased size of those components are also expected. The future electronic products need to be long-lasting and capable of handling a more severe environment [1]. As a result, thermal management and design in the packaging industry has become increasingly important.

An aluminium air-cooled heat sink is one of the most commonly used components for cooling electronic packages. Using such a cooling system is advantageous in that it has a simple maintenance process, more reliability, lower manufacturing cost and no environmental concerns compared to other cooling methods. It has been seen by many researchers that a heat-sink with good geometrical design provides better cooling performance. This means that the optimization process could be an effective design tool for the heat sink. A lot of research work has been conducted in the field of heat sink design/optimization, e.g., in [1]–[7]. It is obvious from the literature that most of the optimization studies are limited to a single objective function whereas, in reality, there are multiple objectives to

be optimized. If the multiobjective design of heat sinks can be achieved, it will surely make a significant impact on electronic packaging technology.

The successful use of the multiobjective evolutionary algorithms (MOEAs) for practical problems has been reported for many years. Using such optimizers is advantageous as they are simple to implement, need no function derivatives, and can deal with almost all kinds of design functions and variables. Moreover, the multiobjective evolutionary optimization process is robust. The most outstanding ability of the multiobjective evolutionary optimizers is that they can explore a Pareto optimum set within one simulation run. They do however have some undesirable drawbacks, which are a lack of consistency and slow convergence rate. As a result, the set of design solutions obtained from using MOEAs is sometimes called an approximate Pareto optimal set. The optimizers are said to be unsuitable for a design problem with highly expensive function evaluation. Therefore, the hybridization of a response surface method (RSM) and the multiobjective evolutionary optimizers has been invented and this approach is found to be very powerful and effective [8], [9].

In this study, the geometrical design of plate-fin heat sinks is carried out using the strength Pareto evolutionary algorithm (SPEA) with and without a response surface method. The design problem is assigned to finding heat sink geometries that minimize heat sink junction temperature and fan pumping power. Design constraints are defined in such a way that the maximum and minimum fin heights are properly limited. The function evaluation is achieved by using the CFD software. The two multiobjective strategies are implemented to explore the Pareto optimal front. From the optimization results obtained, it is seen that the evolutionary multiobjective optimizer with the use of the response surface technique is far superior to the other. The applied multiobjective evolutionary technique is a powerful tool for the design of electronic air-cooled heat sinks.

II. STRENGTH PARETO EVOLUTIONARY ALGORITHM

A multiobjective design problem is posed to find a set of design solutions which minimize multiple objective functions while fulfilling predefined constraints. The objective functions are normally conflicting. For example, in structural optimization, minimizing structural weight usually decreases structural strength while maximizing structural strength results in an increased structural weight. The set of optimum solutions obtained from solving such a problem is called a Pareto optimal set or Pareto front if considered in the objective space. Fig. 1 shows the feasible region and the Pareto front of a particular bi-objective problem on the domain of objective functions.

The basic concept of exploring Pareto optimum points via MOEAs or population-based optimizers is that, on each loop while a new population is created, non-dominated solutions are classified and carried on to the next generation. The term non-dominated solutions defines the local Pareto solutions among the members of the current population and non-dominated solutions from the previous iteration. For the minimization case, two definitions associated with non-domination are given as:

Definition 1: Dominance Given $f_i(\mathbf{x})$ for $i = 1, \dots, m$ are objective functions, if $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ for every

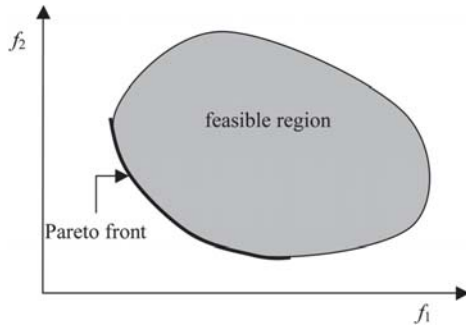
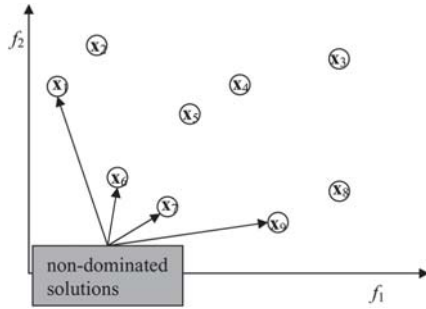
Fig. 1. Pareto front on the domain of objective functions f_1 and f_2 .

Fig. 2. Non-dominated solutions.

$i \in \{1, \dots, m\}$ and there is j such that $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$, then $\mathbf{x}_1$ dominates $\mathbf{x}_2$.

Definition 2: Non-Dominated Solutions (Local Pareto Set)

Given a set of solutions or population $\mathbf{P}$ size N , a solution $\mathbf{x}_e \in \mathbf{P}$ is a non-dominated solution in $\mathbf{P}$ if there does not exist $\mathbf{x} \in \mathbf{P}$ such that $\mathbf{x}$ dominate $\mathbf{x}_e$.

Fig. 2 depicts a plot of 9 design solutions on the domain of objective functions. The non-dominated solutions are $\mathbf{x}_1, \mathbf{x}_6, \mathbf{x}_7$, and $\mathbf{x}_9$.

The multiobjective evolutionary optimizer used to find the Pareto optimum solutions in this research work is SPEA. SPEA is the Pareto approximation method developed by Zitzler and Thiele [10]. The method has been upgraded leading to the second version SPEA2, which is detailed in [11]. SPEA is somewhat based upon genetic algorithm operators. For SPEA2, the search procedure starts with an initial population. An initial external Pareto set or archive is then obtained from sorting the initial population. Fitness values are assigned to the combined members from the current population and external archive based upon the level of domination. The external Pareto set is updated by means of environmental selection. In cases that the number of non-dominated solutions exceeds the predefined size of the external Pareto archive, some solutions will be removed from the external archive by using the truncation operator [also known as the nearest neighborhood method (NNM)]. A number of solutions (equal to the predefined population size) are selected from the updated external Pareto set and put into a mating pool by means of binary tournament selection. The selected solutions are recombined and mutated leading to the new population while objective function values are evaluated. The population and the external Pareto archive are updated repeatedly until the termination criterion is met.

The SPEA method used in this paper is the second version using real numbers as design variables. The real code crossover or recombination is a slight modification from the predecessor [12]. Given that N is the population size and M is the number of design variables, the recombination can be carried out in such a way that, with two randomly selected solutions $\mathbf{x}^1$ and $\mathbf{x}^2$, an offspring $\mathbf{x}$ is obtained by the following strategy

$$x_i = \begin{cases} x_i^1 \text{ or } x_i^2; & P_r < 1/3 \\ x_i^1 + \lambda(x_i^2 - x_i^1); & 1/3 \leq P_r < 2/3 \\ x_i^1 + \text{rand}_1 \cdot (x_i^2 - x_i^1); & P_r \geq 2/3 \end{cases} \quad (1)$$

where $P_r \in [0, 1]$ and $\lambda \in [-0.25, 1.25]$ are uniform random numbers. The process of creating one offspring needs M loops of computation. Before looping, the random number P_r is generated and one of the three recombination strategies in (1) is selected based upon the value of P_r . The random number λ needs to be created if the second strategy is chosen while a uniform random number $\text{rand}_1 \in (-0.25, 1.25)$ needs to be generated in every loop if the third crossover strategy takes place.

The mutation can be carried out by selecting the i th element of the parent solution $\mathbf{x}^P$ at random. The i th element of its child is changed according to

$$x_i = \begin{cases} l_i + \text{rand}_2 \cdot (x_i^P - l_i); & P_r < 0.5 \\ x_i^P + \text{rand}_2 \cdot (u_i - x_i^P); & \text{otherwise} \end{cases} \quad (2)$$

where $\text{rand}_2 \in [-0.25, 1.25]$ is a uniform random number, and l_i and u_i are the lower and upper limits, respectively. It should be noted that the offspring from real code crossover and mutation can exceed the upper and lower limits; therefore, a treatment to keep all offspring inside the bounds is needed. Also, note that the two evolutionary operators are allowed to take place by the given crossover and mutation probabilities.

III. HYBRIDIZATION OF SPEA WITH A RESPONSE SURFACE METHOD

A response surface method is a proven numerical strategy for use in an optimization process. The successful results in using RSM for both single and multiple objective optimizations have been reported worldwide [8], [9], [13]. The basic concept is to exploit numerical curve fitting or interpolation to approximate non-linear design objectives and constraints. The optimization is carried out using the approximate model rather than using the real function evaluation. This means that the problem of time-consuming function evaluation can be alleviated. However, it also has some disadvantages, e.g., inaccurate response surface model may lead evolutionary search to an improper region of design space or even away from the real optima.

The numerical procedure of a coupled SPEA and RSM for an unconstrained case is illustrated in Fig. 3. Coupling RSM with SPEA can be achieved by using SPEA as the main procedure as illustrated in the figure. Initially, a population $\mathbf{P}^k$ is randomly created where k is an iteration number. The external Pareto archive $\mathbf{A}^k$ is found by means of non-dominated sorting. Some design points in the current population are taken to build a response surface model. The response surface model here is based on the surface spline interpolation technique [8], [13], [14]. For the two-objective case, let $\{\mathbf{x}_i\}$ be the set of N solutions from the current population. The approximate function

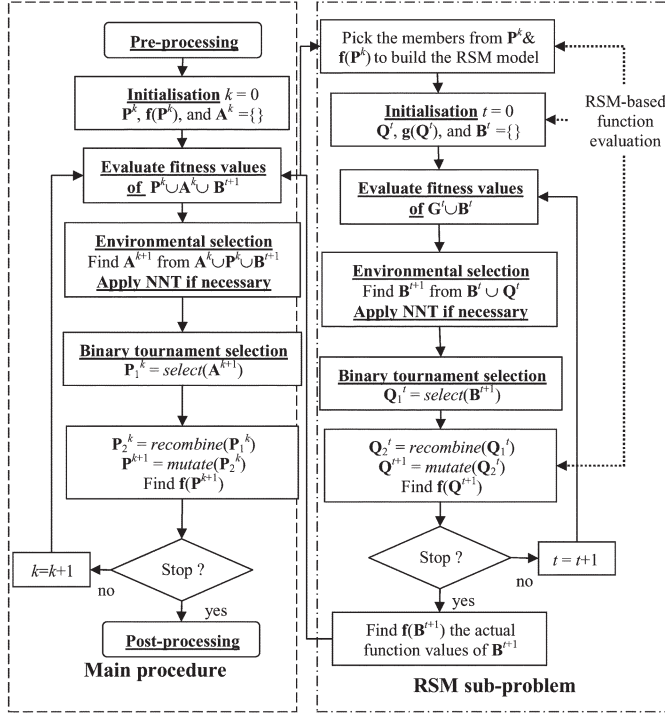


Fig. 3. Flowchart for coupled SPEA and RSM.

values of f_1 and f_2 , written as g_1 and g_2 , respectively, for a design solution $\mathbf{y}$ can be expressed as

$$g_1(\mathbf{y}) = \sum_{i=1}^N p_i \phi(\|\mathbf{x}_i - \mathbf{y}\|) \quad (3)$$

and

$$g_2(\mathbf{y}) = \sum_{i=1}^N q_i \phi(\|\mathbf{x}_i - \mathbf{y}\|) \quad (4)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the vectors of design variables.

The interpolation coefficients p_i can be found from solving the system of linear equations

$$\sum_{j=1}^N p_j \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) = f_1(\mathbf{x}_i); \quad \text{for } i = 1, \dots, N. \quad (5)$$

Equation (5) can be written in the matrix form as

$$\mathbf{K}\mathbf{p} = \mathbf{F}^1 \quad (6)$$

where $K_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and $F_i^1 = f_1(\mathbf{x}_i)$.

For simplicity, the radial basis function is set to be

$$\phi(\|\mathbf{x}_i - \mathbf{x}_j\|) = \|\mathbf{x}_i - \mathbf{x}_j\| = r_{ij} \quad (7)$$

where $\|\mathbf{x}_i - \mathbf{x}_j\| = r_{ij}$ is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The coefficient vector $\mathbf{p}$ can then be determined as

$$\mathbf{p} = \mathbf{K}^\# \mathbf{F}^1 \quad (8)$$

where $\mathbf{K}^\#$ is the pseudo-inverse of matrix $\mathbf{K}$. The pseudo-inverse operator is employed to prevent the searching process

from being prematurely terminated since it is possible that the matrix $\mathbf{K}$ can become singular.

Similarly to the vector $\mathbf{p}$, the coefficient vector $\mathbf{q}$ can be found as

$$\mathbf{q} = \mathbf{K}^\# \mathbf{F}^2 \quad (9)$$

where $F_i^2 = f_2(\mathbf{x}_i)$.

Having built the RSM model, another SPEA sub-optimizer is used to explore the Pareto optimal set using the approximate functions g_1 and g_2 as the objective functions. The non-dominated solutions ($\mathbf{B}^{t+1}$ in Fig. 3) obtained from this stage are taken back to the main optimization procedure. The real function values (f_1 and f_2) of the non-dominated solutions $\mathbf{B}^t$ are evaluated.

The fitness values of the members in the current external archive $\mathbf{A}^k$, the population $\mathbf{P}^k$ and the design points taken from optimizing the RSM model $\mathbf{B}^{t+1}$ are evaluated. The external archive is then updated by operating the environmental selection leading to $\mathbf{A}^{k+1}$. In cases that the number of non-dominated solutions exceeds the archive size, the nearest neighborhood technique is activated. The members in $\mathbf{A}^{k+1}$ selected for the mating pool using the binary tournament selection are recombined and mutated leading to a new population $\mathbf{P}^{k+1}$. This process is repeated until the termination criterion is fulfilled.

The computational steps for SPEA with the use of RSM are as follows.

- 1) Initialization: start with $k = 0$, initial population $\mathbf{P}^k = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, the corresponding objective function values $\mathbf{f}(\mathbf{P}^k)$ and the external Pareto archive $\mathbf{A}^k = \{\}$.
- 2) Chose some members of $\mathbf{P}^k$ and their corresponding objective values $\mathbf{f}_i(k)$ to build a response surface model g_1 and g_2 as in (3)–(9).
- 3) Search for the Pareto front based on the approximate function values $\mathbf{g} = [g_1, g_2]^T$.

3.1 Initialization: start with $t = 0$, initial population $\mathbf{Q}^t = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$, the corresponding objective function values $\mathbf{g}(\mathbf{Q}^t)$ and the external Pareto archive $\mathbf{B}^t = \{\}$.

3.2 Assign the fitness values $\mathbf{G}^t$ to the members in the union set $\mathbf{Q}^t \cup \mathbf{B}^t$.

3.3 Update the Pareto archive by applying the so-called environmental selection to the union set $\mathbf{B}^t \cup \mathbf{Q}^t$ resulting in $\mathbf{B}^{t+1}$.

3.4 If the number of members in $\mathbf{B}^{t+1}$ exceeds the pre-defined archive size, activate NNT to properly remove some of them from the archive.

3.5 Perform a binary tournament selection $\mathbf{Q}_1^t = \text{select}(\mathbf{B}^{t+1})$.

3.6 Recombination: $\mathbf{Q}_2^t = \text{recombine}(\mathbf{Q}_1^t)$.

3.7 Mutation: $\mathbf{Q}^{t+1} = \text{mutate}(\mathbf{Q}_2^t)$.

3.8 Find $\mathbf{g}(\mathbf{Q}^{t+1})$ the objective function values of $\mathbf{Q}^{t+1}$ using (3) and (4).

3.9 Stop and go to step 3.10 if $t = t_{\max}$, otherwise, set $t = t + 1$ and go to step 3.2

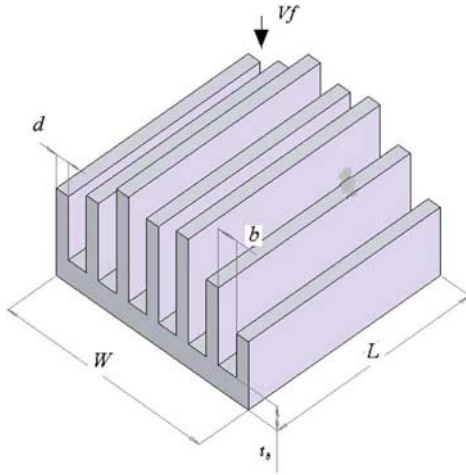


Fig. 4. Plate-fin heat sink.

3.10 Save the non-dominated solutions in $\mathbf{B}^{t+1}$ to step 4.

- 4) Evaluate $\mathbf{f}(\mathbf{B}^{t+1})$ the actual objective function values of the solutions in $\mathbf{B}^{t+1}$.
- 5) Assign the fitness values $\mathbf{F}^k$ to the members in the union set $\mathbf{P}^k \cup \mathbf{A}^k \cup \mathbf{B}^{t+1}$.
- 6) Update the Pareto archive by applying the environmental selection to the union set $\mathbf{P}^k \cup \mathbf{A}^k \cup \mathbf{B}^{t+1}$ resulting in $\mathbf{A}^{k+1}$.
- 7) If the number of members in $\mathbf{A}^{k+1}$ exceeds the predefined archive size, invoke NNT to properly remove some of them from the archive.
- 8) Perform a binary tournament selection $\mathbf{P}_1^k = \text{select}(\mathbf{A}^{k+1})$
- 9) Recombination: $\mathbf{P}_2^k = \text{recombine}(\mathbf{P}_1^k)$
- 10) Mutation $\mathbf{P}^{k+1} = \text{mutate}(\mathbf{P}_2^k)$ and evaluate the objective function values $\mathbf{f}(\mathbf{P}^{k+1})$
- 11) Stop the procedure if $k = k_{\max}$, otherwise, set $k = k + 1$ and go to step 2.

The fitness assignment, environmental selection, truncation (or the nearest neighborhood technique), and binary tournament selection operators are detailed in [11].

IV. MULTIOBJECTIVE DESIGN PROBLEM

The thermal performance and required pumping power of an air-cooled heat sink are dependent on a number of parameters including fin thickness, fin height, number of fins, fin-to-fin spacing, inlet air velocity, and base-plate dimensions. Fig. 4 displays the shape of a typical plate-fin heat sink and some parameters. The fin height distribution can also be assigned as a design variable.

In this work, the multiobjective design problem of the plate-fin heat sink is posed as follows:

$$\begin{aligned} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) &= \{f_1(\mathbf{x}) \quad f_2(\mathbf{x})\} \\ \text{subject to} \\ 10 - H_{\min}/b &\leq 0 \\ H_{\max}/b - 25 &\leq 0 \end{aligned} \quad (10)$$

where

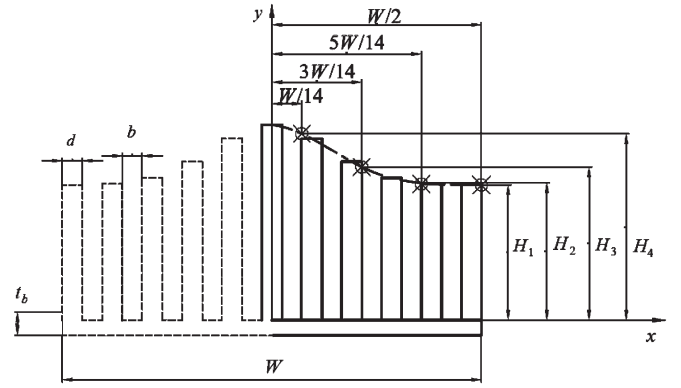


Fig. 5. Design parameters.

f_1 junction temperature f_2 is the fan pumping power;

$H_{\max}$ maximum fin height;

$H_{\min}$ minimum fin height.

The first objective function is a temperature value at the junction of the heat sink and a CPU chip, which can be written as

$$f_1(\mathbf{x}) = T_{jc} = T_a + Q R_{HS} \quad (11)$$

where

T_a ambient air temperature (298 K);

Q heat flux (80 W);

R_{HS} heat sink thermal resistance.

The minimization of this objective value indicates the improved thermal performance of the heat sink. The second objective function affects the fin operating cost, which is rather inevitable in most engineering system design. The fan pumping power can be expressed as

$$f_2(\mathbf{x}) = P_F = \frac{\dot{m}_a \Delta P}{\rho_a} \quad (12)$$

where

P_F fan pumping power;

$\dot{m}_a = \rho_a W L V_f$ air flow rate;

ΔP pressure drop across the heat sink;

ρ_a air density;

V_f inlet air velocity.

Fig. 5 illustrates the design parameters that determine the heat sink cross-sectional area. The vector $\mathbf{x}$ of eight design variables can be detailed as

- $x_1 = d$ fin thickness having the bound as $0.5 \text{ mm} \leq d \leq 3.0 \text{ mm}$;
- $x_2 = n$ fin number having the bound as $5 \leq n \leq 30$;
- $x_3 = t_b$ plate base thickness having the bound as $1.0 \text{ mm} \leq t_b \leq 5.0 \text{ mm}$;
- $x_4 = H_1$ in Fig. 4 having the bound as $2.5 \text{ mm} \leq H_1 \leq 14.0 \text{ mm}$;
- $x_5 = H_2$ in Fig. 4 having the bound as $2.5 \text{ mm} \leq H_2 \leq 14.0 \text{ mm}$;

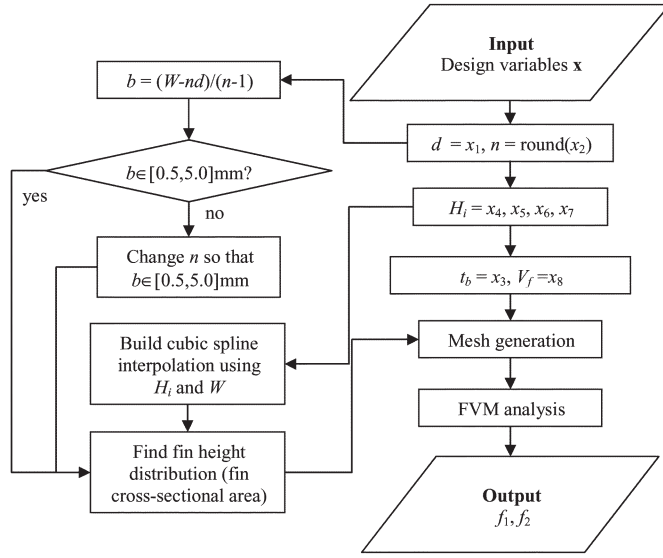


Fig. 6. Flowchart for function evaluation.

$x_6 = H_3$ in Fig. 4 having the bound as
 $2.5 \text{ mm} \leq H_3 \leq 14.0 \text{ mm}$;
 $x_7 = H_4$ in Fig. 4 having the bound as
 $2.5 \text{ mm} \leq H_4 \leq 14.0 \text{ mm}$;
 $x_8 = V_f$ inlet air velocity having the bound as
 $0.5 \text{ m/s} \leq V_f \leq 6 \text{ m/s}$ (see Fig. 4).

The values of H_1, H_2, H_3 , and H_4 control the fin height distribution. The parameters W and L are set to be 60 and 80 mm, respectively.

The procedure of function evaluation is displayed in Fig. 6. Having a vector $\mathbf{x}$ as the input, the parameters d, n, t_b, H_i , and V_f are obtained. The fin-to-fin space b can be computed as $b = (W - nd)/(n - 1)$. If $b \notin [0.5 \text{ mm}, 5.0 \text{ mm}]$, the value of n is changed so that the value of b is bound in the defined range. The value of b has to be limited to enable the mesh generation process. The set of independent variables $\{W/14, 3W/14, 5W/14, W/2\}$ and dependent variables $\{H_1, H_2, H_3, H_4\}$, as shown in Fig. 5, are used to build the cubic spline interpolation [15]. The heights of the n fins are then determined based on this interpolation function. Having all required parameters, the input file for mesh generation is created and run. The analysis process takes place, and the computational results are obtained. The output result for this process is the objective function values f_1 and f_2 .

The thermo-fluid analysis of the forced convection plate-fin heat sink is achieved by using the finite volume method. Fig. 7 displays the solid domain of a particular heat sink being discretized into a number of finite volumes. The grid of the fluid domain is shown in Fig. 8. The finite volume model of the heat sink (both fluid and solid domains) is given in Fig. 9. The assumptions for the analysis are as follows:

- 1) fluid flow is laminar and steady;
- 2) constant thermo-physical material properties of both air and solid;
- 3) uniform inlet air velocity;
- 4) uniform heat flux over entire base plate bottom surface.

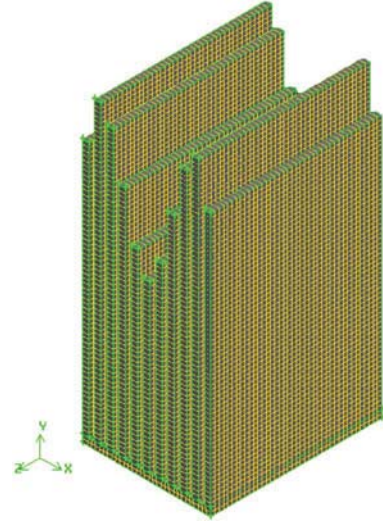


Fig. 7. Mesh generation of a solid domain.

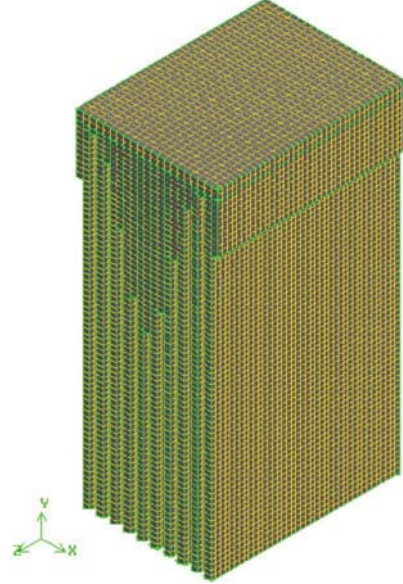


Fig. 8. Mesh generation of a fluid domain.

The heat sink body is made of Aluminium. The physical properties of the solid and fluid are given in Table I.

The SPEA real-code optimizer without the use of RSM (named SPEA) has population size 25 and uses 50 generations for exploring the Pareto front of the optimization problem (10). For SPEA using RSM (named SPEA-RSM), the population size is set to be 25. The Pareto archive $\mathbf{B}^t$ is size 25, which means 25 individuals obtained from optimizing the RSM sub-problem are taken to the main optimization procedure while another 25 individuals are created from the main searching process. In order to have the same number of finite volume analyses as SPEA, SPEA-RSM needs 25 generations. The size of an external Pareto archive $\mathbf{A}^k$ is set to be 80 for both optimization strategies. Both SPEA with and without RSM start searching with the same initial population. The design constraints can be dealt with by using the non-dominated sorting concept presented in [16]. Crossover and mutation probabilities are

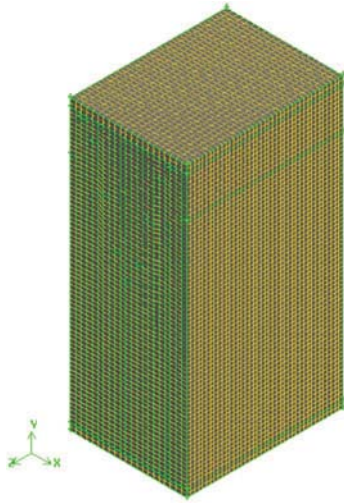


Fig. 9. Finite volume model.

TABLE I
ALUMINIUM AND AIR PROPERTIES

Aluminium properties	Air Properties
Density = 2719.0000 kg/m ³	Density = 1.1770 kg/m ³
Specific heat = 871.0000 J/kg-K	Specific heat = 1006.0000 J/kg-K
Thermal conductivity = 202.0000 W/m-K	Thermal conductivity = 0.0267 W/m-K
	Viscosity = 1.8832 × 10 ⁻⁵ kg/m-s

assigned as 1.0 and 0.5 respectively. According to the predefined population size and the number of generations, the total number of times that the finite volume analysis takes place is equal to 25×50 for both SPEA and SPEA-RSM. For the RSM sub-problem in SPEA-RSM, the number of generations and the population size are set to be 100. It is obvious that the SPEA-RSM needs more computing time to deal with the RSM sub-optimization process but it is insignificant compared to the long period of time needed for each finite volume analysis.

V. DESIGN RESULTS

To simplify explanation, the sets of non-dominated solutions during the optimization process are denoted as: SPEA05-the non-dominated front obtained from using SPEA without RSM after 5 generations, SPEA10-the non-dominated front obtained from using SPEA at the 10th iteration, and so on. Likewise, SPEA-RSM01 means the non-dominated front obtained from using SPEA-RSM at the 1st iteration, and so on. It should also be noted that SPEA10 uses the same number of function evaluations as SPEA-RSM05, which is 250 times.

Fig. 10 shows the evolution of the non-dominated front of the design problem obtained from using SPEA. Note that the non-dominated solutions having the junction temperature higher than 355°K are discarded. The fronts displayed are obtained at the fifth, 10th, 20th, 30th, and 40th generations. It can be seen that the front improved rapidly in the early iterations and changed little in the last iterations. The progress of the

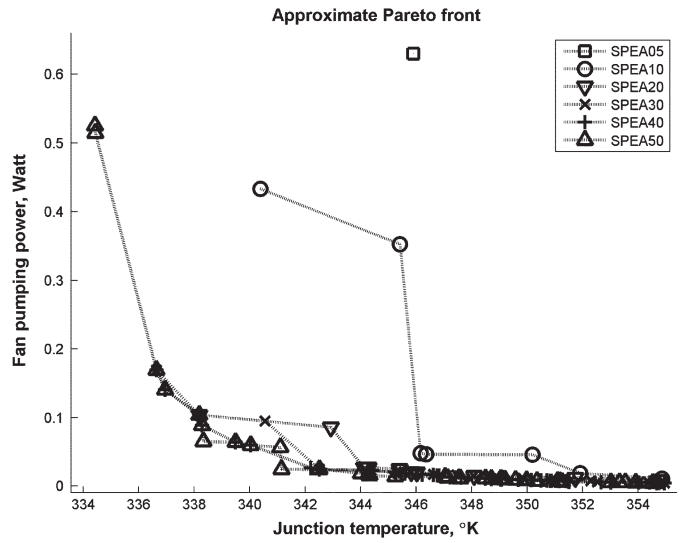


Fig. 10. Progress of SPEA search.

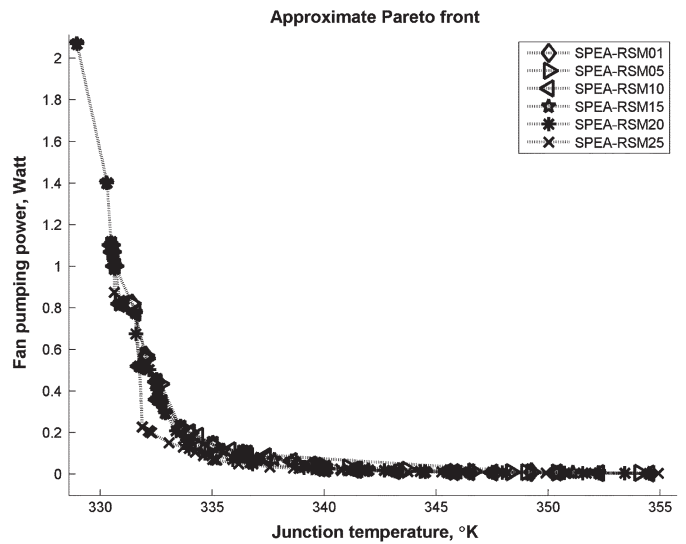


Fig. 11. Progress of SPEA-RSM search.

non-dominated front using SPEA-RSM at the first, fifth, 10th, 15th, and 20th iterations is shown in Fig. 11. It can be observed that the non-dominated front improved to close to the Pareto optimum front right from the first iteration. As the optimization process carried on afterwards, the front only slightly changed.

Performance comparison between SPEA and SPEA-RSM can be made by using the so-called C indicator [17]. The parameter C of a non-dominated front A with respect to front B written as $C(A, B)$ is defined as the number of solutions in B that are dominated or equal to some solution in A divided by the total number of solutions in B . From the definition, if $C(A, B) = 1$, all the solutions in B are dominated by or equal to solutions in A whereas $C(A, B) = 0$ implies that none of the solutions in A cover B . Fig. 12 shows the plots of two non-dominated fronts A and B where the total number of solutions in A and B are 7 and 6, respectively. The C values

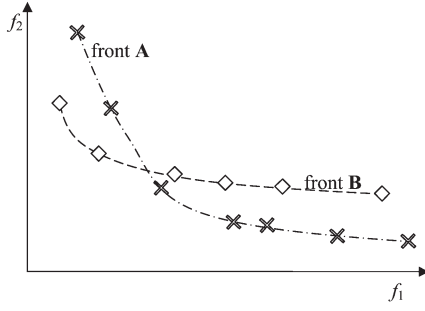


Fig. 12. Comparison of two Pareto fronts.

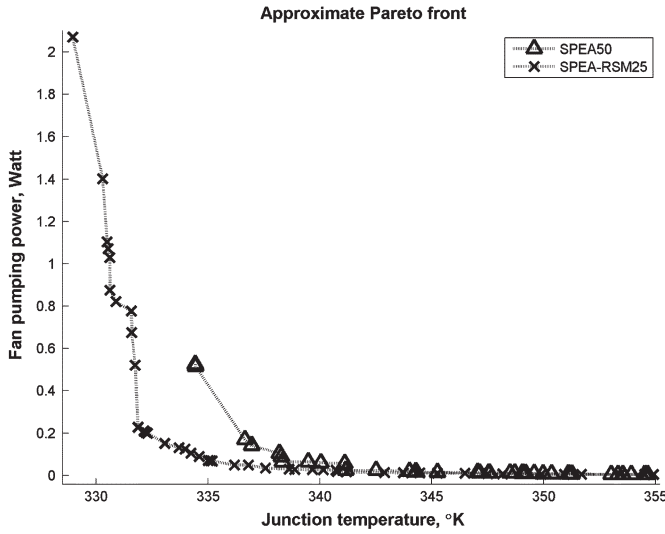


Fig. 13. Comparative Pareto front: SPEA50 versus SPEA-RSM25.

are: $C(A, B) = 4/6$ and $C(B, A) = 2/7$, which means the front **A** is better than front **B**.

The comparison between SPEA with and without the use of RSM is made in Fig. 13. The figure illustrates the approximate Pareto fronts obtained from using SPEA and SPEA-RSM. Both optimization strategies use 1250 function evaluations to form their respective non-dominated fronts. It can be seen that the SPEA-RSM25 front is considerably better than the SPEA50 front. The more obvious superiority of SPEA-RSM over SPEA is displayed in Fig. 14. This figure is a comparison of SPEA50 and SPEA-RSM01 fronts. It should be emphasized that SPEA50 uses 1250 finite volume analyses while it takes merely 50 analyses for SPEA-RSM01. It can be seen that the SPEA-RSM01 front is comparable to the SPEA50 front.

The comparative performance based on the C -indicator of the two optimization strategies is made and shown in Fig. 15. The bar charts given in Fig. 15 are the comparison of C values of the front SPEA50 to the fronts SPEA-RSM01, SPEA-RSM05, SPEA-RSM10, SPEA-RSM15, SPEA-RSM20 and SPEA-RSM25. The first (left-hand side) bar chart is the comparison between SPEA-RSM01 and SPEA50 where C_1 indicates the value of C (SPEA-RSM01, SPEA50) and C_2 is the value of C (SPEA50, SPEA-RSM01). From the C -comparison, the SPEA50 front is slightly better than the SPEA-RSM01 front as C_2 is higher than C_1 . However, as the optimization process continues, the value of C_1 is increasingly higher than that of

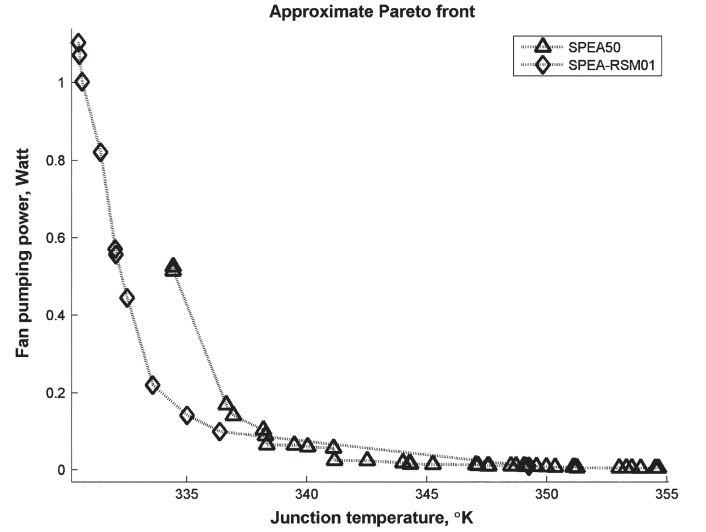


Fig. 14. Comparative Pareto front: SPEA50 versus SPEA-RSM01.

C_2 which means the front obtained from SPEA-RSM becomes increasingly better. At the last generation, the design solutions in the front SPEA-RSM25 are not dominated by any solution from the SPEA50 front as $C(\text{SPEA-RSM25}, \text{SPEA50}) = 1$ and $C(\text{SPEA50}, \text{SPEA-RSM25}) = 0$ in the last bar chart.

The SPEA-RSM25 front is plotted in Fig. 16 where all the solutions are numbered. The heat sink cross-sectional areas of the 60 non-dominated solutions in Fig. 16 are illustrated in Figs. 17 and 18. It is shown that the solutions with lower junction temperatures (min: f_1) have lower fin height fluctuation whereas the solutions with lower fan pumping powers (min: f_2) have higher variation of fin heights.

In order to examine the accuracy of the curve fitting, Fig. 19 displays the root mean square errors (RMSE) of the response surface models during the evolutionary search for both junction temperature and fan pumping power. As detailed in the computational steps, the external Pareto archive of the RSM sub-problem B^{t+1} at step 3.10 was taken to step 4 where the evaluation of actual objective values takes place. The RMSE value here is, therefore, the comparison between the approximate function $g(B^{t+1})$ at step 3.8 and the actual function $f(B^{t+1})$ at step 4. The RMSE of the junction temperature can be computed as

$$\text{RMSE}_1 = \sqrt{\frac{1}{N_A} \sum_{i=1}^{N_A} (f_1(y_i) - g_1(y_i))^2} \quad (13)$$

and the RMSE of the fan pumping power can be computed as

$$\text{RMSE}_2 = \sqrt{\frac{1}{N_A} \sum_{i=1}^{N_A} (f_2(y_i) - g_2(y_i))^2} \quad (14)$$

where $y_i \in B^{t+1}$ and N_A is the total number of design solutions in B^{t+1} .

In Fig. 19, the RMSE of junction temperature is rather high at the early iterations. As the optimization process carries on, the value of RMSE fluctuates but tends to be reduced. It should be noted that the value of junction temperature is commonly in the range of 330 °K–355 °K. Thus, RMSE values of 10 °K–20 °K

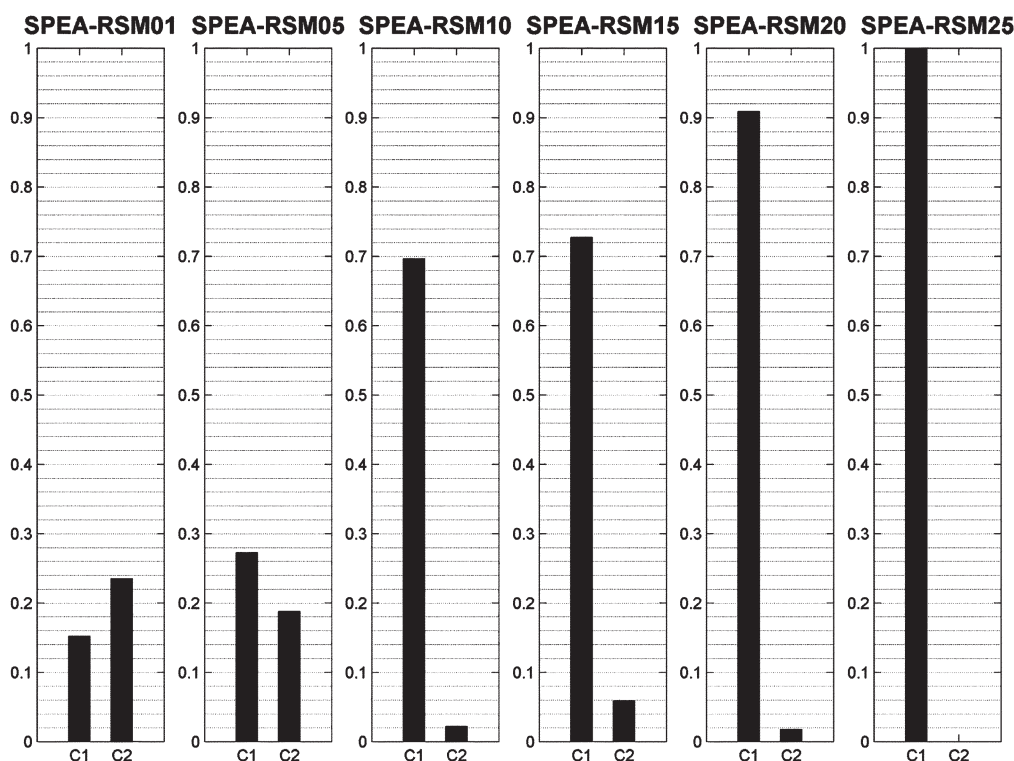


Fig. 15. C-comparison of SPEA50 with SPEA-RSM 01, 05, 10, 15, 20, and 25.

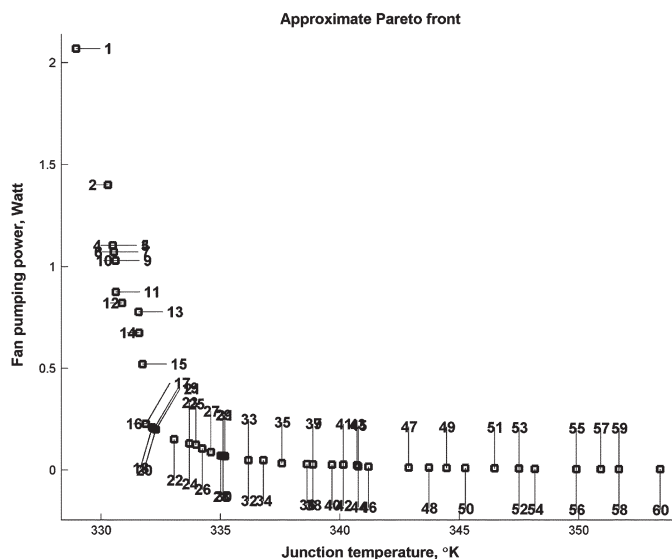


Fig. 16. Pareto front of SPEA-RSM25.

are said to acceptably small. The RSM of the fan pumping power has high errors at the early iterations but it is reduced approaching zero as the optimization process continues. According to Figs. 15 and 18, it can be said that the RSM model is less accurate for the early iterations. This makes the front SPEA50 better than the front SPEA-RSM01. However, as the process continues, the curve fitting becomes more accurate and the non-dominated fronts obtained at the subsequent iterations of SPEA-RSM can surpass the front SPEA50.

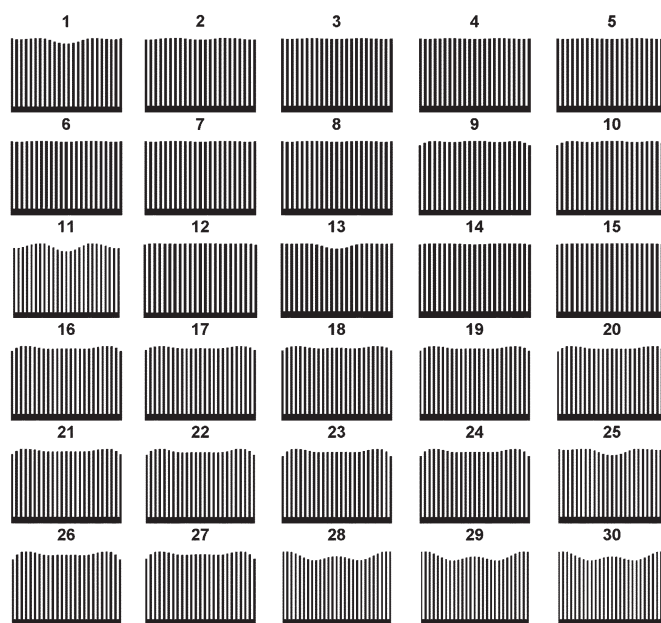


Fig. 17. Heat sink cross-sectional areas of the non-dominated solutions 1–30.

VI. CONCLUSION

The hybridization of real-code strength Pareto evolutionary algorithm with the response surface method is detailed. The use of SPEA with and without the combination of the response surface method for the practical multiobjective design of plate-fin heat sink geometry is demonstrated. From the obtained Pareto optimum results, the linked SPEA and RSM method is far superior to the SPEA method without using RSM. SPEA-RSM is

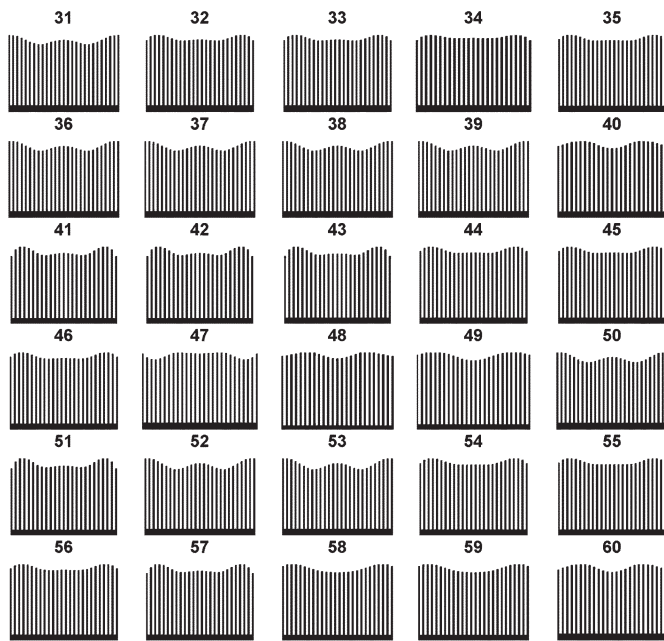


Fig. 18. Heat sink cross-sectional areas of the non-dominated solutions 31–60.

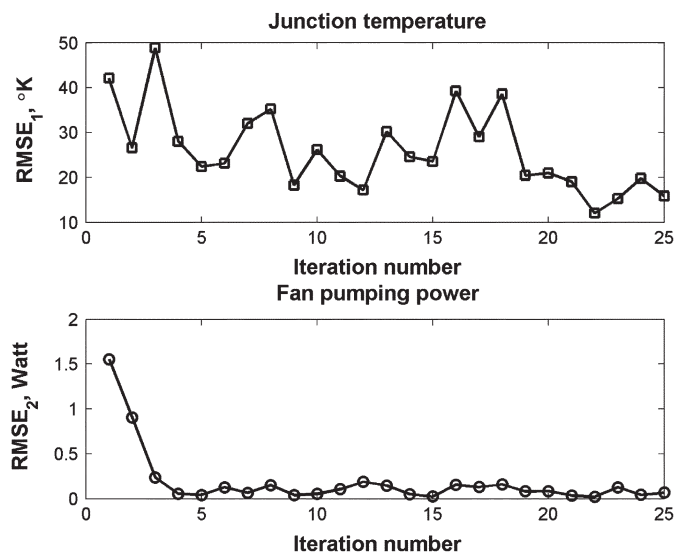


Fig. 19. Root mean square error of f_1 and f_2 .

a powerful practical design tool for multiobjective geometrical design of plate-fin heat sinks. Designers can have a number of optimum heat sinks for decision making within one optimization run. With the use of such a method, all aspects of design variables and functions can be dealt with.

REFERENCES

- [1] W. B. Krueger and A. Bar-Cohen, "Optimal numerical design of forced convection heat sinks," *IEEE Trans. Compon. Packag. Technol.*, vol. 27, no. 2, pp. 417–425, Jun. 2004.
- [2] J. R. Culham and Y. S. Muzychka, "Optimization of plate fin heat sink using entropy generation minimization," *IEEE Trans. Compon. Packag. Technol.*, vol. 24, no. 2, pp. 159–165, Jun. 2001.

- [3] J. A. Visser, D. J. de Kock, and F. D. Conradie, "Minimization of heat sink mass using mathematical optimization," in *Proc. 16th IEEE SEMI-THERM Symp.*, 2000, pp. 252–259.
- [4] A. Bar-Cohen and M. Iyengar, "Least-energy optimization of air-cooled heat sinks for sustainable development," *IEEE Trans. Compon. Packag. Technol.*, vol. 26, no. 1, pp. 16–25, Mar. 2003.
- [5] M. Iyengar and A. Bar-Cohen, "Least-energy optimization of forced convection plate-fin heat sinks," *IEEE Trans. Compon. Packag. Technol.*, vol. 26, no. 1, pp. 62–70, Mar. 2003.
- [6] D. J. de Kock and J. A. Visser, "Trade-off design of extruded heat sinks using mathematical optimization," *ASME J. Electron. Packag.*, vol. 126, pp. 333–341, 2004.
- [7] C. J. Shih and G. C. Liu, "Optimal design methodology of plate-fin heat sinks for electronic cooling using entropy generation strategy," *IEEE Trans. Compon. Packag. Technol.*, vol. 27, no. 3, pp. 551–559, Sep. 2004.
- [8] M. Farina and P. Amato, "Linked interpolation-optimization strategies for multicriteria optimization problems," *Soft Computing*, vol. 9, pp. 54–65, 2005.
- [9] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, "Response surface approximation of Pareto optimal front in multi-objective optimization," *Comput. Methods Appl. Mech. Eng.*, vol. 196, pp. 879–893, 2007.
- [10] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
- [11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2 improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evolutionary Methods Design, Optimiz. Contr.*, Barcelona, Spain, 2002, pp. 95–100.
- [12] H. Muhlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm—Continuous parameter optimization," *Evol. Comput.*, vol. 1, no. 1, pp. 25–49, 1993.
- [13] S. Shan and G. G. Wang, "An efficient Pareto set identification approach for multi-objective optimization on black-box functions," *ASME Trans., J. Mech. Design*, vol. 127, no. 5, pp. 866–874, 2005.
- [14] D. B. McDonald, W. G. Grantham, W. L. Tabor, and M. J. Murphy, "Global and local optimization using radial basis function response surface models," *Appl. Math. Model.*, vol. 31, no. 10, pp. 2095–2110, Oct. 2007.
- [15] G. Lindfield and J. Penny, *Numerical Methods Using MATLAB*. London, U.K.: Ellis Horwood, 1995, pp. 284–294.
- [16] K. Deb, A. Pratap, and T. Meyarivan, "Constrained Test Problems for Multi-Objective Evolutionary Optimization," Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology, Kanpur, India, KanGAL Rep. 200002, 2000.
- [17] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.



Sungkom Srisomporn received the B.Eng. and M.Eng. degrees in mechanical engineering from Khon Kaen University, Khon Kaen, Thailand, in 1998 and 2003, respectively, where he is currently pursuing the Ph.D. degree.

He joined the Faculty of Science and Engineering, Kasetsart University Chalermphrakiat Sakonnakhon Province Campus. His research interests include: finite element analysis, finite volume analysis, and thermal design of electronic devices.



Sujin Bureerat received the B.Eng. degree in mechanical engineering from Khon Kaen University, Khon Kaen, Thailand, in 1992, and the Ph.D. degree in engineering from Manchester University, Manchester, U.K., in 2001.

Currently, he is an Assistant Professor with the Department of Mechanical Engineering, Khon Kaen University. His research interests include: multidisciplinary design/optimization, evolutionary computation, aeroelastic design of aircraft structures, finite element analysis, and experimental

modal analysis.

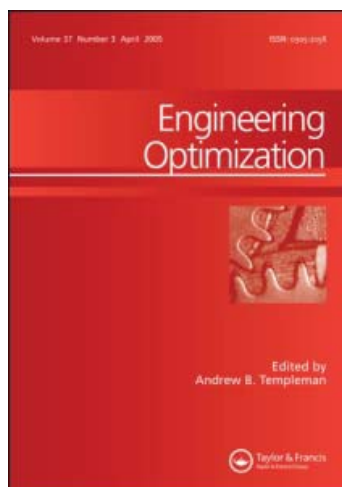
This article was downloaded by: [Bureerat, S.]

On: 23 March 2010

Access details: Access Details: [subscription number 920117359]

Publisher Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Engineering Optimization

Publication details, including instructions for authors and subscription information:

<http://www.informaworld.com/smpp/title~content=t713641621>

Optimum plate-fin heat sinks by using a multi-objective evolutionary algorithm

S. Bureerat ^a; S. Srisomporn ^a

^a Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen, Thailand

First published on: 29 January 2010

To cite this Article Bureerat, S. and Srisomporn, S. (2010) 'Optimum plate-fin heat sinks by using a multi-objective evolutionary algorithm', *Engineering Optimization*, 42: 4, 305 — 323, First published on: 29 January 2010 (iFirst)

To link to this Article: DOI: 10.1080/03052150903143935

URL: <http://dx.doi.org/10.1080/03052150903143935>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Optimum plate-fin heat sinks by using a multi-objective evolutionary algorithm

S. Bureerat* and S. Srisomporn

*Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University,
Khon Kaen, Thailand, 40002*

(Received 31 October 2008; final version received 21 June 2009)

This article demonstrates the practical applications of a multi-objective evolutionary algorithm (MOEA) namely population-based incremental learning (PBIL) for an automated shape optimization of plate-fin heat sinks. The computational procedure of multi-objective PBIL is detailed. The design problem is posed to find heat sink shapes which minimize the junction temperature and fan pumping power while meeting predefined constraints. Three sets of shape design variables used in this study are defined as: vertical straight fins with fin height variation, oblique straight fins with steady fin heights, and oblique straight fins with fin height variation. The optimum results obtained from using the various sets of design variables are illustrated and compared. It can be said that, with this sophisticated design system, efficient and effective design of plate-fin heat sinks is achievable and the best design variables set is the oblique straight fins with fin height variation.

Keywords: heat sink optimum design; multi-objective evolutionary algorithm; finite volume method; shape optimization; computational fluid dynamics

Nomenclature

A	Fin base area.
$\mathbf{B}$	Binary population, binary design variables.
$\bar{b}_j$	Mean value of the j th column of the n_0 selected binary solutions.
b	Fin-to-fin space.
d	Fin thickness.
$f_i, \mathbf{f}$	Objective functions.
$f_i^{\max}$	Maximum value of the i th objective function.
$f_i^{\min}$	Minimum value of the i th objective function.
g_i	Inequality constraints.
h_i	Equality constraints.
H_i	Parameters for determining fin height distribution.
$H_{\max}$	Maximum fin height.

*Corresponding author. Email: sujbur@kku.ac.th

$H_{\min}$	Minimum fin height.
l	Number of probability vectors in Ω .
L	Heat sink length.
L_R	Learning rate.
m	Number of objective functions.
$\dot{m}_a$	Air mass flow rate.
m_s	Mutation shift.
n	Number of design variables.
N	Number of fins.
N_A	External Pareto archive size.
n_b	Number of binary bits for each design solution.
N_P	Number of design solutions in a population.
n_0	Number of randomly selected binary solutions for updating Ω .
Ω	Probability matrix.
Ω_{areto}	External Pareto archive.
P_F	Fan pumping power.
ΔP	Pressure drop across a heat sink.
Q	Heat flux.
$rand$	Uniform random number.
R_{HS}	Heat sink thermal resistance.
S_i, s_i	Parameters controlling fin oblique distribution.
t	Plate base thickness.
T_a	Ambient air temperature.
T_{jc}	Junction temperature.
u_i	Normalised value of the i th objective function.
V_f	Inlet air velocity.
W	Heat sink width.
$\mathbf{x}$	Vector of design variables.
X_i	Oblique parameter of the i th fin.

Greek symbols

ρ_a	Air density.
----------	--------------

Abbreviations

CFD	Computational fluid dynamics.
CPU	Central processing unit.
MOEA	Multi-objective evolutionary algorithm.
MPSO	Multi-objective particle swarm optimization.
PBIL	Population-based incremental learning.
SSM	Response surface method.
SPEA	Strength Pareto evolutionary algorithm.

1 Introduction

High temperature on electronic devices such as a CPU chip is said to be undesirable since it can cause the devices to suffer from heat stress and consequently to have a shorter lifetime than expected. As a result, the use of heat sinks to alleviate heat on such electronic devices has been introduced. Heat sinks are increasingly important due to the increase of power densities in

electronic packages as well as the reduction of their size and weight (Craig *et al.* 1999). There have been several cooling mechanisms for heat sinks and one of them is an air-cooled system. Using such an air-cooled system is advantageous since it has a simpler maintenance process, more reliability, lower manufacturing cost and no environmental concerns when compared with other cooling methods (Srisomporn and Bureerat 2008).

The application of optimization process for heat sink design has been studied for many years (Craig *et al.* 1999, Yu *et al.* 2002, Bar-Cohen and Iyengar 2003, Iyengar and Bar-Cohen 2003, Khan *et al.* 2004, Rueger and Bar-Cohen 2004, Shih and Liu 2004). Some work is concerned with single-objective optimization. Some studies deal with multi-objective optimization of heat sinks using approximation models such as a response surface method (RSM) (Park and Moon 2005, Chiang and Chang 2006), riging model (Park *et al.* 2006), and grey-fuzzy logic based on orthogonal arrays (Chiang *et al.* 2006), while function evaluation is carried out by means of experiment or CFD analysis. In most cases, the design variables are merely sizing parameters of heat sink fins and, moreover, the application of an automated optimum design is limited. The recent work (Srisomporn and Bureerat 2008) has presented the use of hybridisation of SPEA (Nitzler *et al.* 2002) and RSM to solve the multi-objective design of a vertical straight plate-fin heat sink having fin height fluctuation. A more advanced design system is proposed, *i.e.* the combination of optimization codes written in MATLAB and commercial CFD software. It has been found that using the hybrid of SPEA and RSM is superior to purely using the SPEA method.

The work in this article is an extension of that presented by Srisomporn and Bureerat (2008). The use of PBIL in combination with finite volume commercial software for shape optimization of plate-fin heat sinks is presented. The more complicated heat sink shapes are studied. Three shape design variable sets including: vertical straight fins with fin height variation, oblique straight fins with steady fin heights, and oblique straight fins with fin height variation, are defined and used for the design demonstration. The optimum results obtained from using the various sets of design variables are illustrated and compared. It can be said that, with this more sophisticated design system, efficient and effective design of plate-fin heat sinks can be achieved. The best design variable set is the oblique fins with the variation of fin heights.

Multi-objective population-based incremental learning

Multi-objective evolutionary algorithms have been well established since the last decade. Using this kind of optimizer is said to be advantageous as they are simple to use, robust, and derivative-free. The methods can deal with almost all kinds of optimum design problems and the most outstanding capability of MOEAs is that they can search for a set of Pareto optimal solutions within one optimization run. They have some drawbacks, however, which are a lack of consistency and a low convergence rate since their searching procedures greatly rely on randomization. Some well-known MOEAs are, for example, strength Pareto evolutionary algorithm and multi-objective particle swarm optimization (MPSO) (Deb-Sierra and Coello Coello 2006). The recent development of the estimation of distribution method for multi-objective design problems is PBIL (Bureerat and Sriwornamas 2007, Anyakam and Bureerat 2007), which is found to be as efficient and effective as the other classical MOEAs.

A multi-objective design problem is posed to find a set of design solutions which minimize multiple objective functions simultaneously while fulfilling design constraints. Such a problem can be written as:

Find $\mathbf{x}$ such that

$$\text{Min: } \mathbf{f} = \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\} \quad (1)$$

Subject to

$$g_i(\mathbf{x}) \leq 0$$

$$h_i(\mathbf{x}) = 0$$

The objective functions are normally conflicting with each other. The set of optimum solutions obtained from solving such a problem is called a Pareto optimal set or Pareto front if considered in the objective space as shown in Figure 1.

In the past, most optimizers were developed to deal with a single objective problem. When dealing with a multi-objective problem, additional numerical schemes such as the weighted sum technique are used to convert the problem to become a single objective case. The weak point in using such numerical techniques is that the optimization method needs to be performed as many times as the number of Pareto optimal solutions required. This difficulty can, however, be avoided when using MOEAs.

The basic concept of exploring Pareto optimum points via MOEA is that, at each generation while a new population is created, non-dominated solutions are classified and carried on to the next generation. The term non-dominated solutions define the local Pareto solutions among the members in the current population during evolutionary search. The non-dominated solutions obtained at each generation are saved to the so-called external Pareto archive.

The MOEA used in this article is the multi-objective version of PBIL, which has an optimization procedure based upon binary searching space. Rather than keeping all binary solutions or a binary population as with genetic algorithms, the population in PBIL is represented by the probability vector of having 1 of each bit position of the binary strings. Figure 2 shows three probability vectors producing three binary populations where each row vector of the populations represents a design solution. It is shown that one probability vector can form a variety of populations.

When dealing with multi-objective optimization, more probability vectors are used in order to obtain a more diverse population; therefore, it is called a probability matrix instead. The PBIL search starts with an (empty) external Pareto archive and an initial probability matrix whose elements are full of 0.5. Each row of the matrix is a probability vector used to create a sub-population. Let N_P be the number of design solutions in a population, l be the total number of probability vectors, and n_b be the number of binary bits for each design solution. The probability matrix, therefore, has the size of $l \times n_b$ where each row produces N_P/l design solutions as one sub-population. Each design variable is represented by a binary string with n_b/n elements. Figure 3

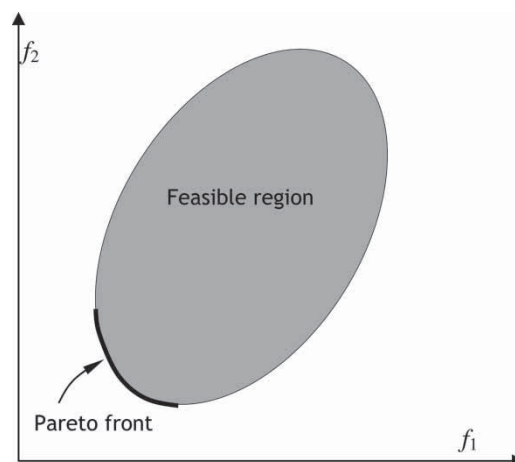


Figure 1. Pareto front on the objective function domain.

population 1				population 2				population 3			
0	0	1	1	1	1	1	0	0	0	0	1
1	1	1	0	0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1	1	0	0	1
1	0	0	0	0	0	0	1	0	1	0	1
Probability Vectors											
[0.5, 0.5, 0.5, 0.5]				[0.5, 0.5, 0.5, 0.5]				[0.25, 0.5, 0, 0.75]			

Figure 2. Probability vectors and their corresponding populations.

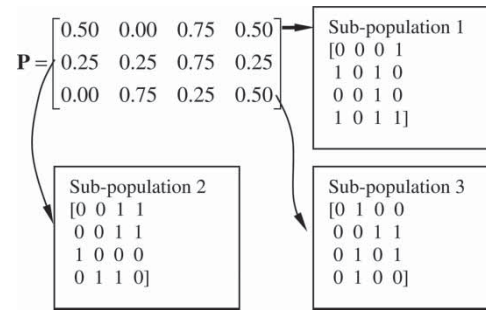


Figure 3. Probability matrix and sub-populations.

displays a 3×4 probability matrix and their corresponding population ($N_P = 12$), which is the union set of three sub-populations according to the three rows of the probability matrix.

Having generated the current population and evaluated their corresponding objective values, the external Pareto archive is filled with the non-dominated members of the union set of the current population and the old external Pareto archive. If the number of the new non-dominated solutions is larger than the predefined archive size (N_A), a Pareto archiving technique called the normal line method (Anyakam and Bureerat 2007) is activated to remove some solutions from the archive.

On each generation, the probability matrix is updated in such a way that $n_0 < N_A$ binary solutions are randomly selected from the current Pareto archive to modify the i th row of the probability matrix. The i th row of the probability matrix is updated using the relation

$$P_{i,j}^{new} = P_{i,j}^{old} (1 - L_R) + \bar{b}_j L_R \quad (2)$$

where L_R is called the learning rate, and $\bar{b}_j$ is the mean value of the j th column of the n_0 selected binary solutions. In this work, the learning rate is set as

$$L_R = 0.5 + rand \cdot (+0.1 \text{ or } -0.1) \quad (3)$$

where $rand \in [0, 1]$ is a uniform random number. It is also useful to afterwards apply mutation to the i th row of the probability matrix as

$$P_{i,j}^{new} = P_{i,j}^{old} (1 - m_s) + rand(0 \text{ or } 1) \cdot m_s \quad (4)$$

where m_s is the amount of shift used in the mutation. Similarly to genetic algorithms, the mutation operator is allowed to take place by a given mutation probability.

The updating process is completed when all rows of the probability matrix are changed. The probability matrix is updated and the external Pareto archive is improved iteratively until a termination condition is met.

The procedure for multi-objective PBIL can be written as:

1. Initialization: probability matrix $\mathbf{p} = [\mathbf{p}_i]_{l \times n_b}$, external Pareto archive $\mathbf{pareto} = \{\}$.
2. Generate a binary population $\mathbf{B}$ from $\mathbf{p}$.
3. Decode the binary population to be $\mathbf{x}_{n \times N_p}$ and find the objective values $\mathbf{f}_{m \times N_p}$.
4. Update $\mathbf{pareto}$ by replacing it with the non-dominated solutions of a union set.

$\mathbf{pareto} \cup \mathbf{x}$

5. If the number of members in $\mathbf{pareto}$ exceeds the predefined archive size N_A , remove some of them by using the normal line technique.
6. If the termination criterion is fulfilled, stop the procedure. Otherwise, go to step 7.
7. Update $\mathbf{p}$.
 - 7.1 For $i = 1$ to l .
 - 7.2 Select n_0 binary solutions from $\mathbf{pareto}$ randomly.
 - 7.3 Update the i th row of $\mathbf{p}$ by using (2) and (3).
 - 7.4 Generate $rand \in [0, 1]$ a uniform random number.
 - 7.5 If $rand <$ the predefined mutation probability, update the i th row of $\mathbf{p}$ using (4).
 - 7.6 Next i .
8. Go to step 2.

For the procedure of the normal line archiving technique, initially, all of the current non-dominated solutions are normalized as:

$$u_i = \frac{(f_i - f_i^{\min})}{(f_i^{\max} - f_i^{\min})} \quad (5)$$

where $f_i^{\min}$ is the minimum value of the i th objective function sorted from the current non-dominated solutions, and $f_i^{\max}$ is the maximum value of the i th objective function.

The numerical procedure in cases of bi-objective optimization is illustrated in Figure 4 (Panayakam and Bureerat 2007). The points that give the minimum values of each objective are taken as the anchor points. The line connecting between the two anchor points is called the utopia line while N_a reference points are evenly spaced on the utopia line. The normal lines are the lines normal to the utopia line, and pass the reference points as shown. A non-dominated point is saved to the archive if it is the closest point to a particular normal line. The non-dominated point is iteratively selected until the external Pareto set is full. Six selected non-dominated points are bounded by a dashed rectangle as shown in Figure 4.

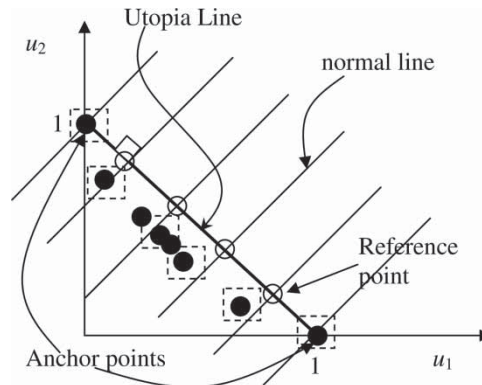


Figure 4. Illustration of the normal line technique.

The capability of PBIL in exploring Pareto fronts has been tested and compared with several established MOEAs such as the second version of SPEA (Bureerat and Sriworamas 2007). It has been found that PBIL is the best in providing population diversity and it is overall the best performer.

□ □ Design problems

It is obvious from the literature that two of the most important design objectives are minimizing heat at the heat sink base and minimizing cooling power. The design problem presented in this article has two objective functions which are junction temperature and fan pumping power. Note that the objective functions, constraints, and some other parameters are taken from the work by Srisomporn and Bureerat (2008). The bi-objective optimization problem can be written as:

$$\min_{\square} \mathbf{f}(\square) = \{f_1(\square) \ f_2(\square)\} \quad (\square)$$

subject to

$$10 - H_{\min}/b \leq 0$$

$$H_{\max}/b - 25 \leq 0$$

$$f_1 - 355 \leq 0$$

where f_1 is a junction temperature ($^{\circ}\square$), f_2 is a fan pumping power ($\square$ att).

The first objective function is a junction temperature between the heat sink and a CP□ chip, which can be expressed as

$$f_1(\square) = T_{jc} = T_a + Q R_{HS}. \quad (7)$$

The minimization of the junction temperature results in the improved thermal performance of the heat sink. The fan pumping power, which affects the fan operating cost, can be expressed as:

$$f_2(\square) = P_F = \frac{\dot{m}_a \Delta P}{\rho_a} \quad (8)$$

where $\dot{m}_a = \rho_a A V_f$ is an air mass flow rate.

An air-cooled plate-fin heat sink is chosen for this design case study. The thermo-fluid analysis of the forced convection plate-fin heat sink is achieved by using the finite volume method. The assumptions for the analysis are as follows:

- Fluid flow being laminar and steady
- Constant material thermo-physical of both air and solid
- □niform inlet air velocity
- □niform heat flux through entire base plate bottom surface.

The heat sink body is made of aluminium. The physical properties of the solid and fluid are given in Table 1.

□ □ □ Encoding/decoding design variables

The bi-objective design problem (□) will be solved with three different sets of design variables, which will be named as follows:

Table 1. Aluminium and air properties.

Aluminium properties	Air Properties
Density = 2719.0000 kg/m ³	Density = 1.1770 kg/m ³
Specific heat = 871.0000 J/kg·K	Specific heat = 1005.0000 J/kg·K
Thermal conductivity = 202.0000 W/m·K	Thermal conductivity = 0.026 W/m·K
	Viscosity = 1.8832 × 10 ⁻⁵ kg/m·s

F1: vertical straight fins with fin height variation.

F2: oblique straight fins with steady fin heights.

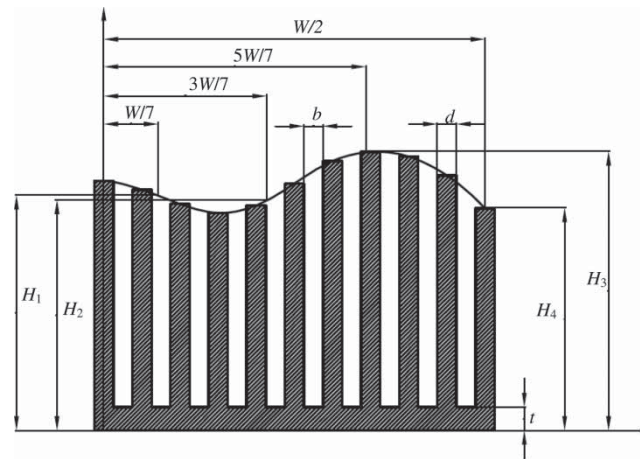
F3: oblique straight fins with fin height variation.

The first design variables set **F1** is the same as that used by Srisomporn and Bureerat (2008). The vector of design variables has eight elements, which can be detailed as:

- $x_1 = d$, fin thickness having the bound as $0.5 \text{ mm} \leq d \leq 3.0 \text{ mm}$.
- $x_2 = N$, fin number having the bound as $5 \leq N \leq 30$.
- $x_3 = t$, plate base thickness having the bound as $1.0 \text{ mm} \leq t \leq 5.0 \text{ mm}$.
- $x_4 = H_1$, in Figure 5 having the bound as $25 \text{ mm} \leq H_1 \leq 50 \text{ mm}$.
- $x_5 = H_2$, in Figure 5 having the bound as $25 \text{ mm} \leq H_2 \leq 50 \text{ mm}$.
- $x_6 = H_3$, in Figure 5 having the bound as $25 \text{ mm} \leq H_3 \leq 50 \text{ mm}$.
- $x_7 = H_4$, in Figure 5 having the bound as $25 \text{ mm} \leq H_4 \leq 50 \text{ mm}$.
- $x_8 = V_f$, inlet air velocity having the bound as $0.5 \text{ m/s} \leq V_f \leq 6 \text{ m/s}$.

Figure 5 displays a half cross-sectional area of a particular heat sink generated by the **F1** design variables. The parameters W and L (Figure 6) are set to be 60 mm and 80 mm respectively. The values of H_1 , H_2 , H_3 and H_4 control the fin height distribution. The computational steps for function evaluation of this set of design variables can be given as follows:

1. Input binary codes **B**.
2. Decode **B** leading to a vector of real number design variables **x**.
3. Assign $d = x_1$, $N = \text{round}(x_2)$, $t = x_3$, $H_1 = x_4$, $H_2 = x_5$, $H_3 = x_6$, $H_4 = x_7$, $x_8 = V_f$.
4. Find the fin-to-fin space $b = (W - Nd)/(N - 1)$.

Figure 5. Design parameters of **F1** problem.

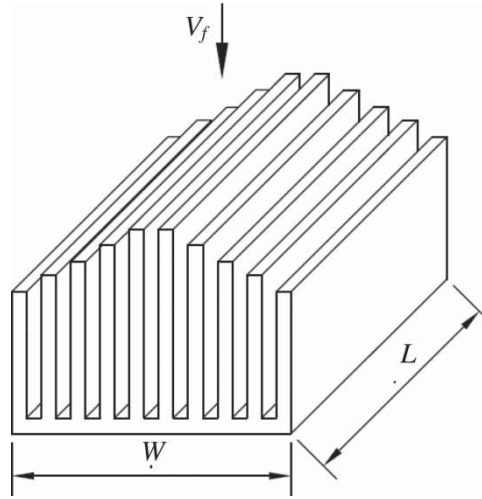


Figure 6. Straight plate-fin heat sink with fin height variation.

5. If b is too small or too large ($b \notin [0.5, 5]$ mm), decrease or increase the value of N until the value of b is in the proper range.
6. Interpolate the independent variables $[W/7, 3W/7, 5W/7, W/2]$ and dependent variables $[H_1, H_2, H_3, H_4]$ using cubic spline interpolation (Gindfield and Penny 1985).
7. Generate the fin cross-sectional area according to all the parameters previously determined.
8. Create an input file for finite volume analysis.
9. Finite volume analysis
10. Read the output file and compute the values of f_1 and f_2 to enter the optimization process.

The function $\text{round}(x)$ will replace x with its nearest integer. In the fifth step, b needs to be bounded in the certain range (here, $b \in [0.5 \text{ mm}, 5.0 \text{ mm}]$) to enable the grid generation process of the commercial software.

The second set of design variables, **F2**, determines oblique fins with steady fin heights. Using an oblique straight plate-fin heat sink is advantageous since it has a greater surface area compared to a vertical plate-fin heat sink with the same base area and height (Lin et al. 2005). In order to make an automated optimization process dealing with binary codes and the oblique fin shape, a more complicated numerical scheme is required. The difficulty usually arises at the stage of grid generation of the commercial CFD software. Figure 7 shows two particular fins of an **F2** heat sink. To make a legitimate heat sink, the condition (9) has to be met ensuring that there is an adequately big gap between the fins for fluid flow.

$$X_{i+1} \geq X_i \quad (9)$$

The vector of **F2** design variables has 8 elements as $d = x_1$, $N = \text{round}(x_2)$, $t = x_3$, $S_1 = x_4$, $S_2 = x_5$, $S_3 = x_6$, $S_4 = x_7$, and $x_8 = V_f$. The base area is the same as the case of **F1**, $W = 60 \text{ mm}$ and $L = 80 \text{ mm}$ while the fin height is set to be $H = 40 \text{ mm}$. The values of S_1 , S_2 , S_3 , and S_4 are used to determine the distribution of the values X_i of the N fins. With the condition (9), it leads to new design constraints:

$$S_1 \leq S_2 \leq S_3 \leq S_4. \quad (10)$$

These constraints however could cause some difficulty in the optimization process of BFO. To avoid adding such constraints to the design problem, the values of S_1 , S_2 , S_3 and S_4 are

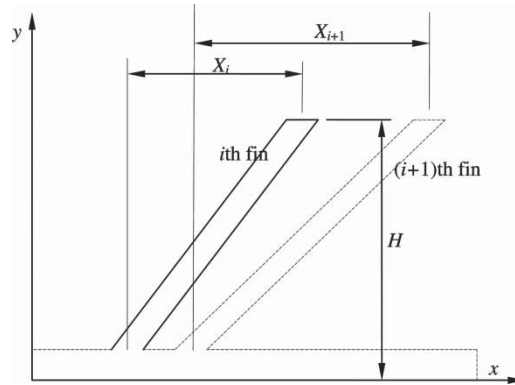


Figure 7. Oblique plate-fin heat sink.

modified as

$$\begin{aligned}
 s_1 &= \frac{S_1}{S_1 + S_2 + S_3 + S_4} S_4 \\
 s_2 &= \frac{S_1 + S_2}{S_1 + S_2 + S_3 + S_4} S_4 \\
 s_3 &= \frac{S_1 + S_2 + S_3}{S_1 + S_2 + S_3 + S_4} S_4 \\
 s_4 &= S_4
 \end{aligned} \tag{11}$$

It is clear that, by using (11), the constraints (10) are always fulfilled. The bounds of the parameters controlling the fin oblique distribution are assigned as $S_1, S_2, S_3 \in [0, 100]$ mm and $S_4 \in [0, 20]$ mm. Apart from that, the problem of grid generation takes place when the gap between the i th and $(i + 1)$ th fins are too small as shown in Figure 8. This problem can be dealt with by reducing the i th fin height as displayed in Figure 9. This means that, in the **F2** design case, varied fin heights can be obtained although steady fin heights are expected. The computational steps for function evaluation of the **F2** design variables are as follows:

1. Input binary codes **B**.
2. Decode **B** leading to a vector of real number design variables **x**.

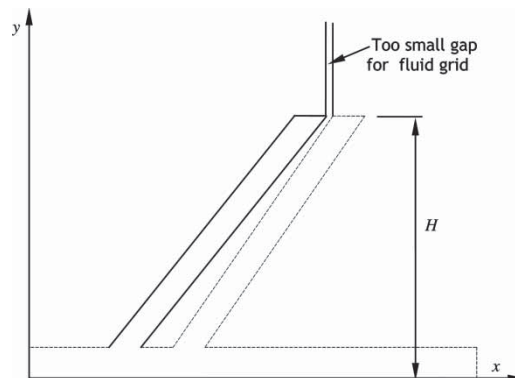


Figure 8. Oblique fins that lead to small gap for mesh generation.

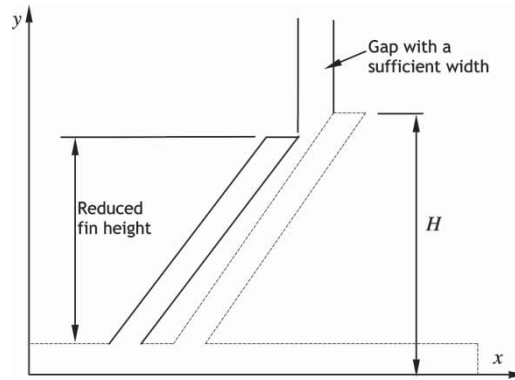


Figure 10 Shortened i th fin leading to a sufficiently wide space for mesh generation.

3. Assign $d = x_1$, $N = \text{round}(x_2)$, $t = x_3$, $S_1 = x_4$, $S_2 = x_5$, $S_3 = x_6$, $S_4 = x_6$, $x_8 = V_f$.
4. Find the fin-to-fin space $b = (W - Nd)/(N - 1)$.
5. If b is too small or too large ($b \notin [0.5, 5]$ mm), decrease or increase the value of N until the value of b is in the proper range.
6. Find s_1, s_2, s_3 and s_4 using (11).
7. Interpolate the independent variables $[W/7, 3W/7, 5W/7, W/2]$ and dependent variables $[s_1, s_2, s_3, s_4]$ using cubic spline interpolation.
8. Find X_i , $i = 1, \dots, N$.
 - 1 Trim the fins for grid generation.
 - 1 For $j = (N - 1)$ to the middle fin.
 - 2 If the gap between the j th and $(j + 1)$ th fin is too small, reduce the height of the j th fin until the gap is adequately large.
 - 3 $\text{end } j$.
10. Make symmetric fin heights and symmetric oblique distribution
11. Generate the fin cross-sectional area according to all the parameters previously determined.
12. Create an input file for finite volume analysis
13. Finite volume analysis
14. Read the output file and compute the values of f_1 and f_2 to enter the optimization procedure.

Figure 10 displays the solid model of a particular oblique fin heat sink being discretized into a number of finite volumes whereas the grid of its fluid model is shown in Figure 11. It can be seen that the fin heights are slightly varied.

The third set of design variables is the combination of **F1** and **F2**. The input design parameters consist of d , N , t , V_f , H_1 , H_2 , H_3 , H_4 , S_1 , S_2 , S_3 , and S_4 . The process of function evaluation starts with decoding binary codes to be the real design parameters. The number of fins are then determined. The fin height distribution is calculated and the heights of the N fins are obtained. Afterwards, the fin oblique distribution is calculated and the trim process is activated to reduce the height of some fins enabling the automatic mesh generation of the software. After the input file has been created, the analysis takes place, the output results are read, and the objective function values are computed.

The NSGA-II multi-objective optimizer is employed to solve each multi-objective optimization problem simultaneously. In each optimization run, the non-dominated solutions in the last updated external Pareto archive are taken as the approximate Pareto front of the problem. The population size is set to be $N_P = 25$ while the total number of generations is set to be 50. The external Pareto archive size is set to be $N_A = 100$, and the number of binary bits for each design variable is 20.

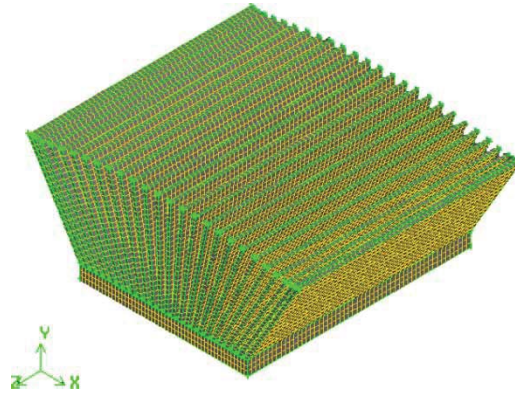


Figure 10. Oblique plate-fin heat sink: solid model.

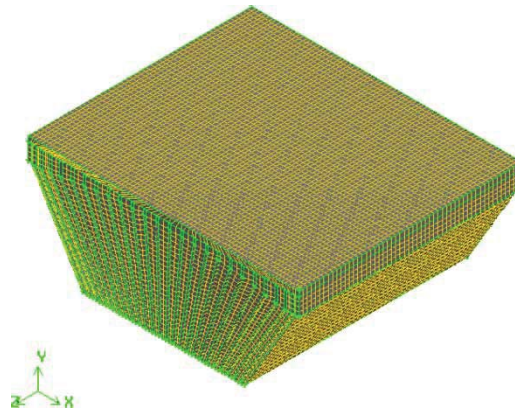


Figure 11. Oblique plate-fin heat sink: fluid model.

The mutation probability is set to be 0.05 while the mutation shift is $m_s = 0.2$. The non-dominated sorting scheme for constrained optimization proposed by Deb (2001) is employed. It should be noted that SGA with real codes (Srisomporn and Bureerat 2008) and PSO with real codes (Ceyes-Sierra and Oello 2006) are also employed to solve the problems six times for each design problem. The crossover and mutation rates for SGA are set as 1.0 and 0.5 respectively while, for PSO, the starting inertia weight, ending inertia weight, cognitive learning factor, and social learning factor are assigned as 0.5, 0.01, 0.5 and 0.5 respectively.

There are totally $3 \times 3 \times 6$ non-dominated fronts from the six runs of the three multi-objective evolutionary optimizers used to solve the three design problems. The comparative performance of BI, SGA and PSO is achieved by means of C -comparison (Zitler 2000). The box-plots of C values comparing six pairs of optimization strategies are illustrated in Figure 12. It can be seen that SGA and BI significantly outperform PSO. SGA and BI are said to be equally good for the case of F1 and F2 (SGA is slightly better for F2) while BI is obviously superior to SGA in cases of the F3 design problem. It should be noted that the predefined parameters used for each multi-objective evolutionary algorithm usually affect its searching performance. Since

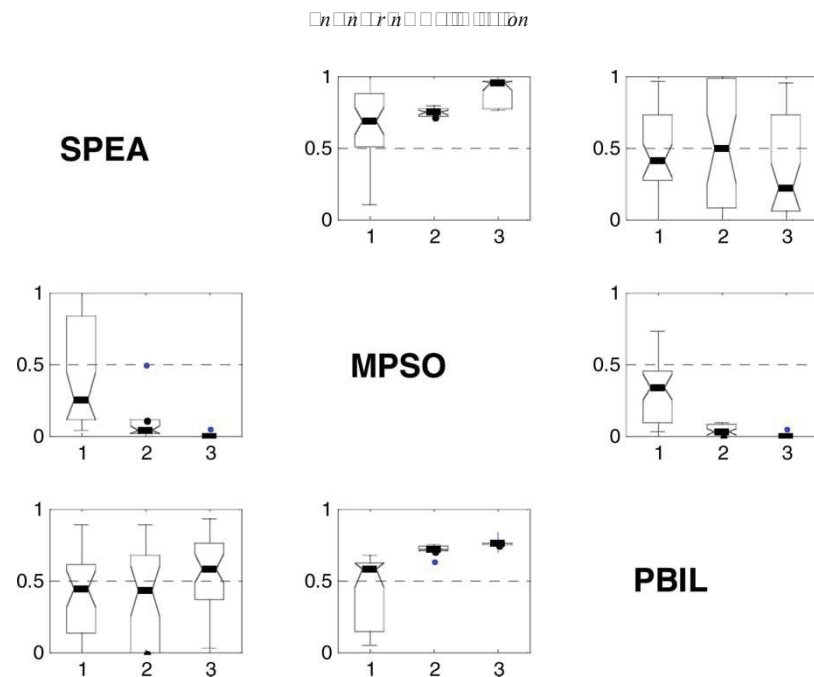


Figure 12. Comparative performance of the algorithms based on the C indicator.

the overall best method is PBIL, only the results obtained from using PBIL are therefore chosen to present.

The Pareto fronts obtained from solving the F1 problem six times are shown in Figure 13. It can be seen that the results from using PBIL are quite consistent as they can reach almost the same

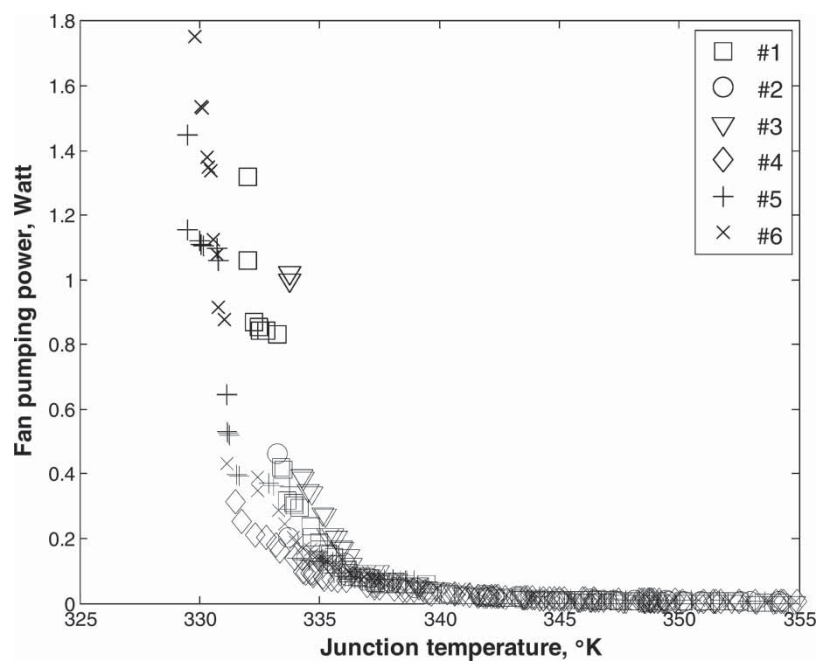


Figure 13. Six approximate Pareto fronts of F1.

frontier for all simulations. The cross-sectional areas of the heat sinks from the best front (sorted by using the C indicator) are shown in Figures 14 and 15. From the figures, all of the heat sinks tend to have the certain number of fins. Most of the heat sinks have slightly varied fin heights while some of them have obvious fin height fluctuation.

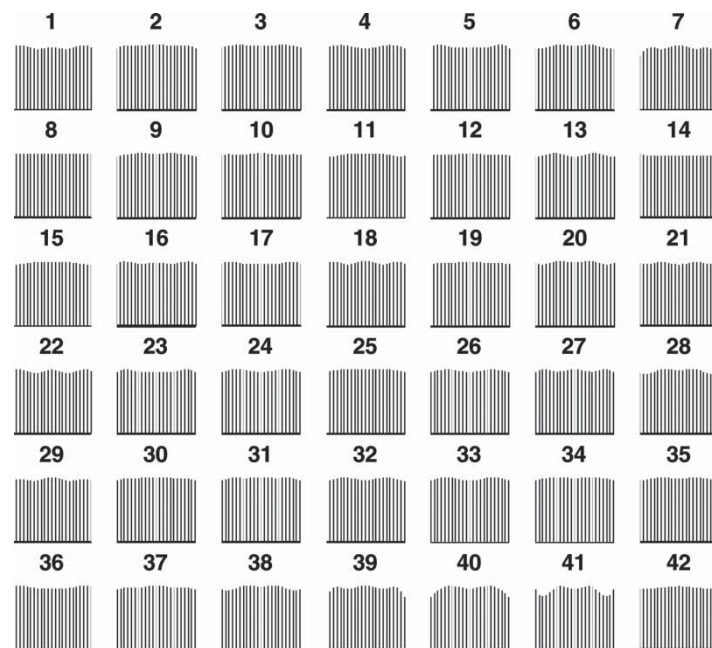


Figure 14. Cross-sectional areas of the heat sinks on the best Pareto front of F1: I.

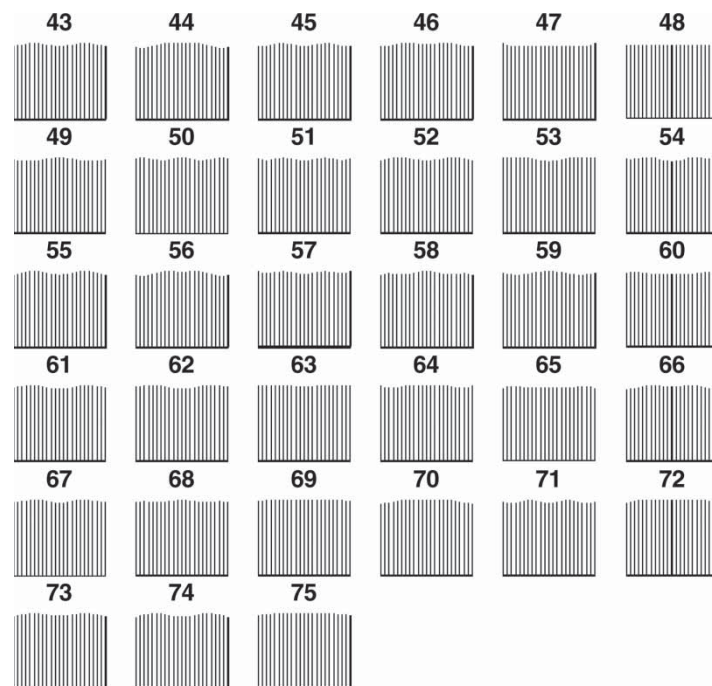


Figure 15. Cross-sectional areas of the heat sinks on the best Pareto front of F1: II.

The simulated Pareto fronts of **F2** are displayed in Figure 16. The searching performance of PBIL for this case can be considered as being consistent, although PBIL cannot reach the same frontier for all simulations. The heat sinks' cross-sectional areas of the best Pareto front are illustrated in Figures 17 and 18. It can be seen that all of the heat sinks have varied fin orientations and most of the heat sinks have slightly varied fin heights.

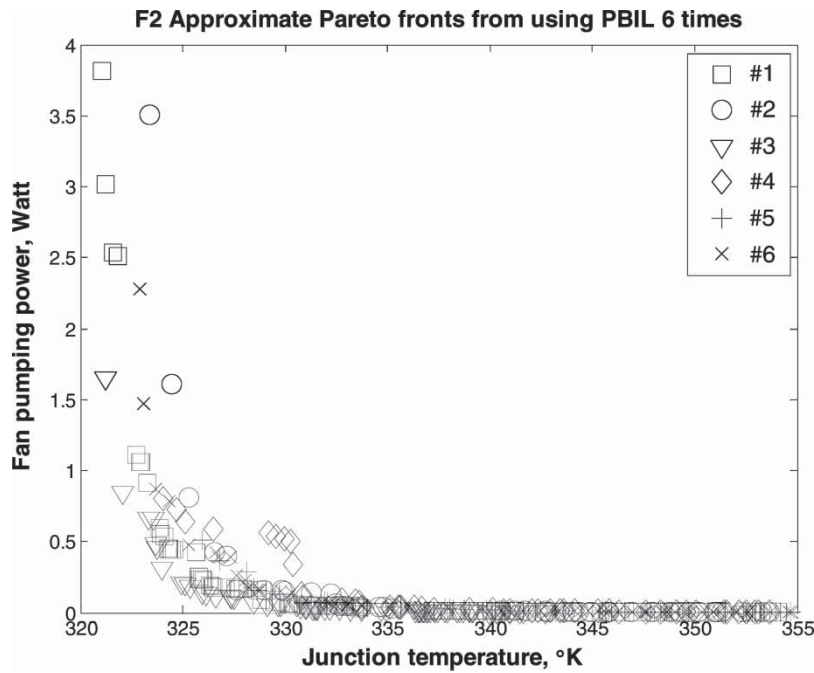


Figure 16. Simulated Pareto fronts of **F2**.

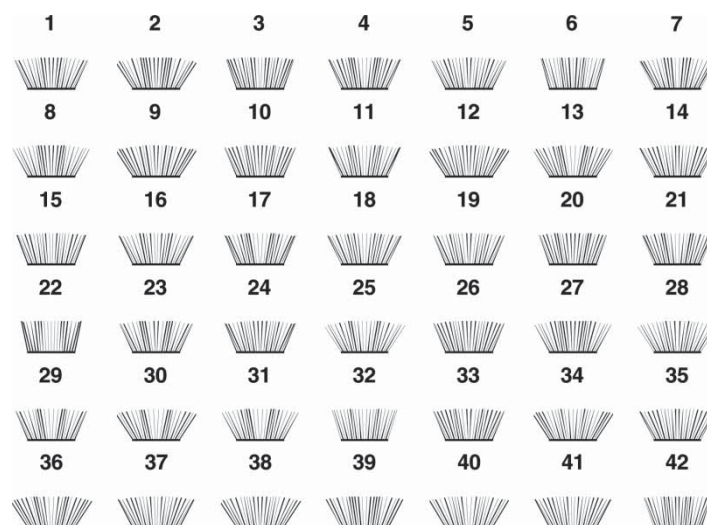


Figure 17. Cross-sectional areas of the heat sink on the best Pareto front of **F2**: I.

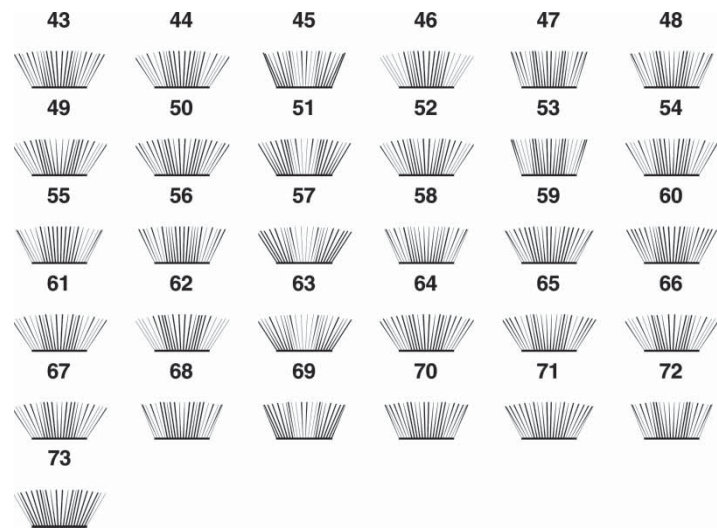


Figure 18. Cross-sectional areas of the heat sink on the best Pareto front of F2: II.

Since Pareto fronts from solving the F3 problem sometimes are plotted in Figure 1. The searching performance of PBIL for solving this design problem is said to be less consistent compared to the previous design problems. The cross-sectional areas of the heat sinks of the best front are illustrated in Figures 20 and 21. It can be seen that the heat sinks have fins with variation of fin orientations and heights.

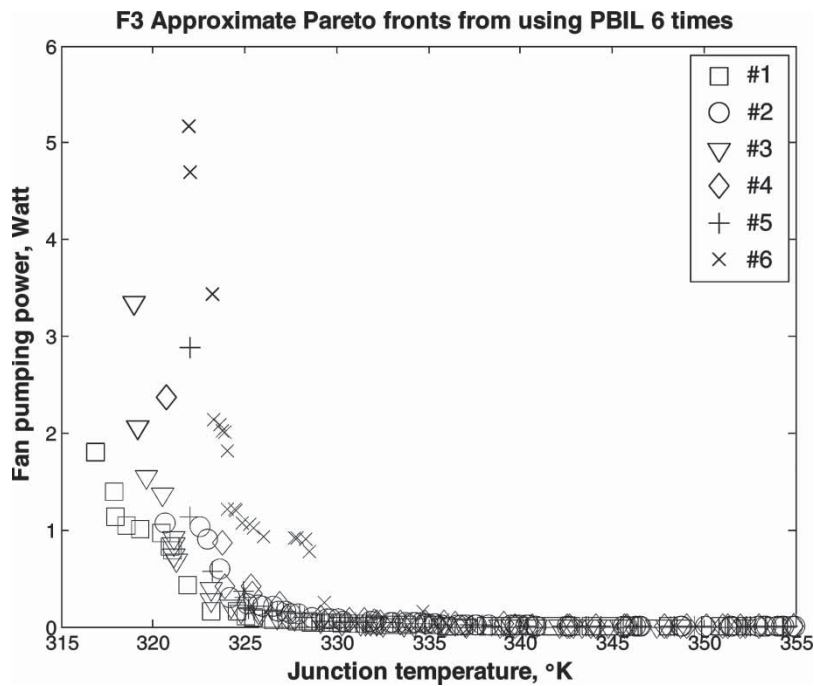


Figure 19. Six approximate Pareto fronts of F3.

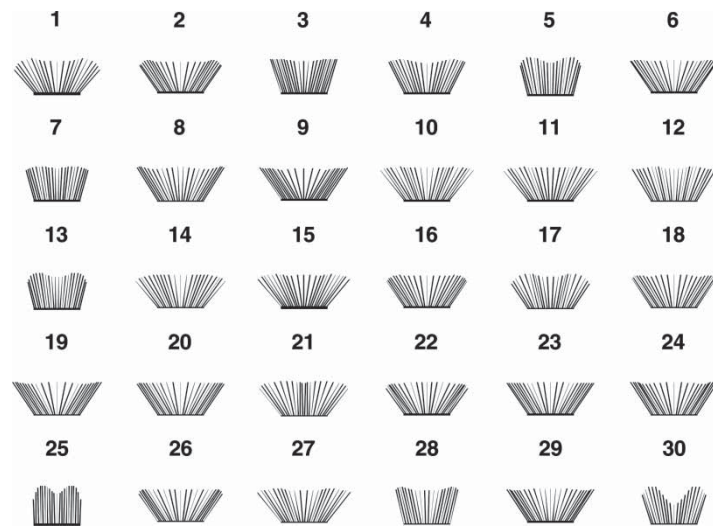


Figure 20. Cross-sectional areas of the heat sink on the best Pareto front of **F3**: I.

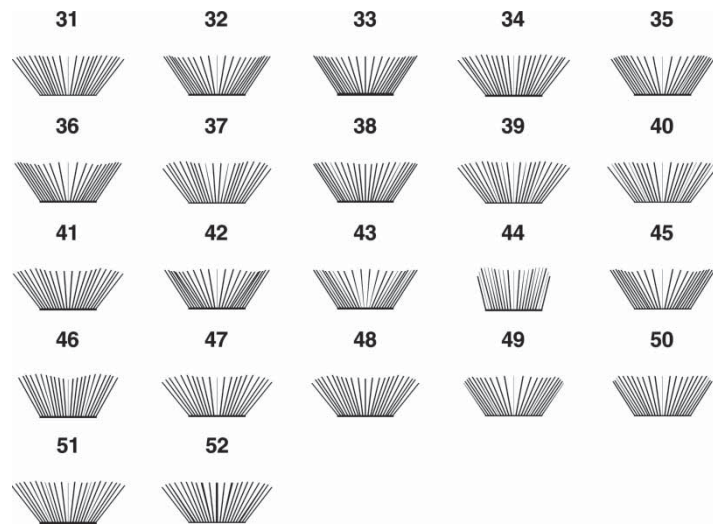
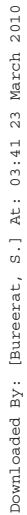


Figure 21. Cross-sectional areas of the heat sink on the best Pareto front of **F3**: II.

The best fronts of the problems **F1**, **F2** and **F3** are plotted together in Figure 22 to make a comparison. The diamond markers are objective values (obtained from CFD analysis) of a typical real heat sink shown in Figure 23 where $V_f \in \{2, 3, 4, 5, 6\}$ m/s, $t = 8$ mm, $d = 2$ mm, $H = 32$ mm, $N = 21$, $W = 63$ mm, and $L = 80$ mm. It can be seen that the best front that totally dominated the rest is the Pareto front of the **F3** problem while the second and third bests are the front of **F2** and **F1** respectively. This implies that using the wider design space such as **F3** results in the better heat sinks. The real heat sink, which has a lower fin height but slightly larger base area, is inferior to the other design solutions. This can be said that the proposed automated optimization process is an effective and efficient design tool although it is not reasonable to compare the heat sinks from solving **F1**, **F2** and **F3** with the real heat sink directly.



The computational procedure of multi-objective BBO is detailed. The three sets of design variables that determine the heat sink shape are proposed. The concept and computational steps for function evaluation of each design variables set are given. With the use of BBO, a set of Pareto optimal solutions can be obtained within one optimization run. The proposed design problems are difficult or even impossible to be solved by gradient-based optimization methods since the design problems have a mix of integer and real design variables, and function derivatives are unavailable. From

the Pareto optimal results obtained, the design variables set that give the best results is the **F3** set as expected while **F2** is better than **F1**. The searching performance of **EBI** in dealing with the three multi-objective design problems is said to be acceptable based on its consistency. By using this sophisticated design system, unconventional design problems can be posed and solved and this may lead the thermal design of heat sinks to another level.

□ □ □ □ □ □ □ □ □ □ □ □ □ □

The authors are grateful for the support from the Thailand Research Fund (TRF). Any thanks are also directed to our colleague, Peter Barr, for making this article more readable.

□ □ □ □ □ □ □ □

- Bar-Elmehrik, A. and Iyengar, R., 2003. Least-energy optimization of air-cooled heat sinks for sustainable development. *Journal of Electronic Packaging*, 125, 16–25.
- Bureerat, S. and Sriwornamas, S., 2007. Population-based incremental learning for multi-objective optimization. *Engineering Optimization*, 39, 223–232.
- Chiang, T.T. and Chang, F.H., 2006. Application of response surface methodology in the parametric optimization of a pin-fin type heat sink. *International Journal of Numerical and Analytical Methods in Engineering*, 33, 836–845.
- Chiang, T.T., Chang, F.H., and Tsai, T.H., 2006. Optimum design parameters of pin-fin heat sink using the grey-fuzzy logic based on the orthogonal arrays. *International Journal of Numerical and Analytical Methods in Engineering*, 33, 744–752.
- Craig, R., de Groot, D., and Bauchb, R., 1999. Minimization of heat sink mass using CFD and mathematical optimization. *Journal of Electronic Packaging*, 121, 143–147.
- Deb, K., Pratap, A., and Invarivan, T., 2001. Constrained test problems for multiobjective evolutionary optimization. *Evolutionary Computation*, 9, 183–201, 284–288.
- Iyengar, R. and Bar-Elmehrik, A., 2003. Least-energy optimization of forced convection plate-fin heat sinks. *Journal of Electronic Packaging*, 125, 62–70.
- Kanyakam, S. and Bureerat, S., 2007. Passive vibration suppression of a walking tractor handlebar structure using multi-objective EBI. *Proceedings of the 25th International Conference on Mechanical Engineering*, Singapore, 1–3, 4162–4163.
- Khan, A.A., Gulham, R., and Ivanovich, R., 2004. Optimization of pin-fin heat sinks using entropy generation minimization. *Journal of Electronic Packaging*, 126, 267–274.
- Rueger, B. and Bar-Elmehrik, A., 2004. Optimal numerical design of forced convection heat sinks. *Journal of Electronic Packaging*, 126, 417–425.
- Sin, S., Chuang, F.S., and Hou, A., 2005. Experimental study of the heat sink assembly with oblique straight fins. *Journal of Electronic Packaging*, 127, 500–505.
- Windfield, R. and Penny, R., 1995. *Numerical Methods for Engineers*. London: Ellis Horwood, 284–284.
- Park, K. and Moon, S., 2005. Optimal design of heat exchangers using the progressive quadratic response surface model. *International Journal of Numerical and Analytical Methods in Engineering*, 48, 2126–2133.
- Park, K., Huh, K., and Kim, K., 2006. The application of the CFD and Taguchi method to an optimization of heat sink. *International Journal of Numerical and Analytical Methods in Engineering*, 49, 343–347.
- Eyes-Sierra, R. and Coelho Coelho, A., 2006. Multiobjective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Numerical and Analytical Methods in Engineering*, 2, 287–308.
- Yu, K., Choi, D., and Kim, S., 2002. Numerical optimization of the thermal performance of a microchannel heat sink. *International Journal of Numerical and Analytical Methods in Engineering*, 45, 2823–2827.
- Shih, K. and Liu, K., 2004. Optimal design methodology of plate-fin heat sinks for electronic cooling using entropy generation strategy. *Journal of Electronic Packaging*, 126, 551–555.
- Srisomporn, S. and Bureerat, S., 2008. Geometrical design of plate-fin heat sinks using hybridization of GA and PSO. *Journal of Electronic Packaging*, 130, 351–360.
- Fritter, K., Deb, K., and Thiele, K., 2000. Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8, 173–185.
- Fritter, K., Baumanns, K., and Thiele, K., 2002. SGA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: K. Giannakoglou (eds.), *Evolutionary Computation for Engineering Design*. Barcelona, Spain. International Center for Numerical Method in Engineering, 15–100.

Optimal Design of a Pin-Fin Heat Sink Using a Surrogate-Assisted Multiobjective Evolutionary Algorithm

Siwadol Kanyakam^{1,a}, Sujin Bureerat^{2,b}

^{1,2}Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand.

^aohsiwa@yahoo.com, ^bsujbur@kku.ac.th

Keywords: Pin-fin heat sink, Multiobjective evolutionary algorithm, Population-based incremental learning, Surrogate model, Kriging model, Computational fluid dynamics

Abstract. In this work, performance enhancement of a multiobjective evolutionary algorithm (MOEA) by integrating a surrogate model to the design process is presented. The MOEA used in this work is multiobjective population-based incremental learning (PBIL). The bi-objective design problem of a pin-fin heat sink (PFHS) is posed to minimize junction temperature and fan pumping power while meeting design constraints. A Kriging (KRG) model is used for improving the performance of PBIL. The training points for constructing a surrogate KRG model are sampled by means of a Latin hypercube sampling (LHS) technique. It is shown that hybridization of PBIL and KRG can enhance the search performance of PBIL.

Introduction

A heat sink is a well-known component used in a cooling system of electronic devices to control operational temperature in order to keep the device away from thermal failure. Based on such a reason, the heat generated at the junction of both heat sink and device (called junction temperature, T_j) becomes one of the most important heat sink design parameters. Moreover, to reduce operation cost, fan pumping power, P_{fan} , also needs to be minimized. This implies that, in designing a heat sink, multiple design objectives should be taken into consideration.

Recently, research work on heat sink optimal design by using evolutionary algorithms (EAs) has been presented using both single and multiple objective function design problems e.g. in [1-5]. Our previous work [5] shows that using MOEAs for multiobjective design of a PFHS is superior to the classical design approach. However, it is also found that the optimization process is time-consuming due to expensive function evaluations from performing finite volume analysis (FVA); therefore, performance enhancement of the MOEA design process is required.

This work is aimed to enhance a multiobjective evolutionary optimizer performance by incorporating evolutionary search with a surrogate model. This design problem is posed to find geometrical parameters of a PFHS to minimize its junction temperature and fan pumping power. Population based incremental learning is chosen as the optimizer since it gives the best results as shown in [1]. The surrogate model employed in this work is the Kriging (KRG) method or a Gaussian process model [6]. The results show that the performance of a surrogate-assisted PBIL can be compared to non-dominated solutions obtained from running of PBIL with a larger number of function evaluations, and without using a surrogate model.

Multiobjective Evolutionary Algorithms (MOEAs)

Engineering design problems usually have multiple design objectives that are often conflict with each other. A multiobjective optimization problem can be detailed as follows:

Find a set of variables $\mathbf{x}$ such that

minimize $\mathbf{f} = \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}$

subject to

$g_i(\mathbf{x}) \leq 0$ (inequality constraints)

$h_i(\mathbf{x}) = 0$ (equality constraints)

where $\mathbf{x} \in \mathbb{R}^n$ is a vector of n design variables, and k objective functions $f_i(x)$ are intended to be minimized simultaneously. In the multiobjective problem, $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\}^T$ is an objective function vector that contains objective functions of $k \geq 2$. A variable vector $\mathbf{x}'$ represents a Pareto optimal solution (or non-dominated solution) if it does not exist any $\mathbf{x}$ in a feasible region such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i = 1, \dots, m$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ for at least one index of j . A set of non-dominated solutions obtained from the multiobjective problem is called a Pareto optimal set. If it is plotted in the objective domain it is called a Pareto or non-dominated front as illustrated in Fig. 1 for the case of bi-objective.

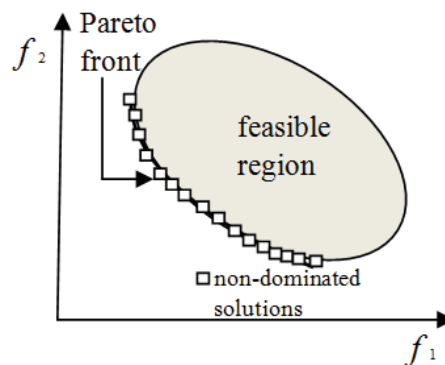


Fig. 1 Pareto optimal set

Population-Based Incremental Learning (PBIL). The PBIL method is based on binary search space as proposed by Baluja [7] in 1994. The search procedure starts with an initial probability vector (P_i) in which each element in the vector determines the probability of having '1' on each a column of a binary population given that each row in the population is one design solution. Samples of a probability vector are given in Fig. 2. It is shown that one probability vector can result in a variety of populations. This feature of PBIL helps maintain population diversity in multiobjective optimization.

Population 1	Population 2	Population 3
0 0 1 0,	0 1 0 1,	1 1 0 1
1 0 0 1,	1 1 1 0,	0 0 0 1
1 1 1 0,	0 0 0 1,	1 1 0 1
0 1 0 1,	0 0 0 0,	0 1 1 0
Probability vectors		
[0.5, 0.5, 0.5, 0.5] [0.25, 0.5, 0.25, 0.5] [0.5, 0.75, 0.25, 0.75]		

Fig. 2 Probability vectors and their corresponding populations

Having obtained a probability vector from the binary population, the best binary design solution is then used to modify the probability vector using the following relationship.

$$P_i^{new} = P_i^{old} (1 - L_R) + b_i L_R$$

where b_i is the element of best binary solution, and L_R is learning rate which herein is defined as:

$$L_R = 0.5 + rand \cdot (+0.1 \text{ or } -0.1).$$

The number $rand \in [0,1]$ is a uniform random number.

For multiobjective PBIL, several probability vectors which will be called a probability matrix are used instead of only one vector as with the single objective case [8]. The search procedure starts with an initial probability matrix, and an empty Pareto archive for collecting non-dominated solutions. A binary population can then be obtained from the union set of sub-populations created from each row of the probability matrix. The non-dominated solutions are obtained by sorting the combination of the non-dominated solutions from the previous generation and the members in the current population. In case that the number of non-dominated solutions exceeds the predefined archive size, a normal line method [1] is employed to remove some solutions from the Pareto archive. The process is repeated until a stopping criterion is fulfilled. More details of multiobjective PBIL can be seen in [1, 8].

Hybridization of PBIL and KRG

In this work, the hybridization of PBIL and KRG is detailed. Initially, the LHS technique is used to generate training points (in this work, 50 points were created.). The KRG model is then constructed by means of the maximum likelihood method [6]. The PBIL is then used to find Pareto solutions while function evaluations are predicted using the KRG model. Having obtained the final solutions, the actual functions are found using computational fluid dynamic (CFD) analysis. Afterwards, design solutions from using the surrogate model are taken as initial solutions for the main PBIL search procedure. The PBIL algorithm is then operated for a few iterations to improve the approximate Pareto front. The flowchart for this hybrid approach is illustrated in Fig. 3.

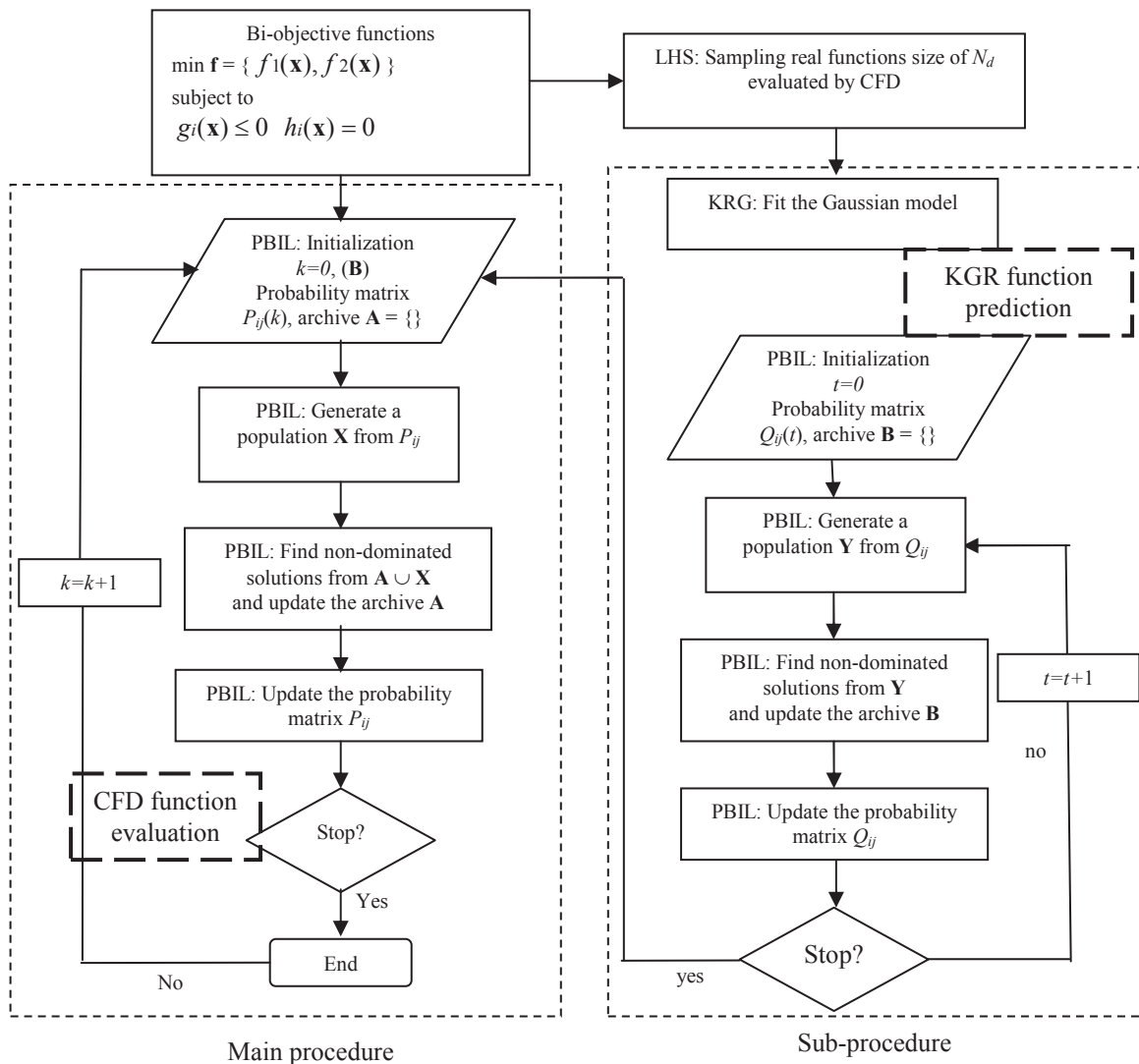


Fig. 3 Flowchart for hybrid PBIL and KRG

Design Problem

Pin-Fin Heat Sink Model. A PFHS physical model with square base (w_{hs}) is given in Fig. 4. Parameters defining a heat sink geometry include base length (w_{hs}), base thickness t_b , fin width (w_{fin}), and number of fins (n_f). The fins heights are varied based on the 16 input parameters (H_1-H_{16}). Figure 5 displays 7×7 control points on the fin base where the 16 fin height parameters are distributed symmetrically. The fin height can be determined by using radial-basis function interpolation [5]. A fin-to-fin space or fin stitch (s_{fin}) is equi-spaced. The heat sink is cooled by a fan of V_f air velocity (m/s) which is installed on the top of a heat sink. A uniform heat load (Q) generated at the junction of heat sink and the electronic device is set to be 120 watt. Materials and air properties and the assumptions of heat-flow in this simulation are detailed in [5].

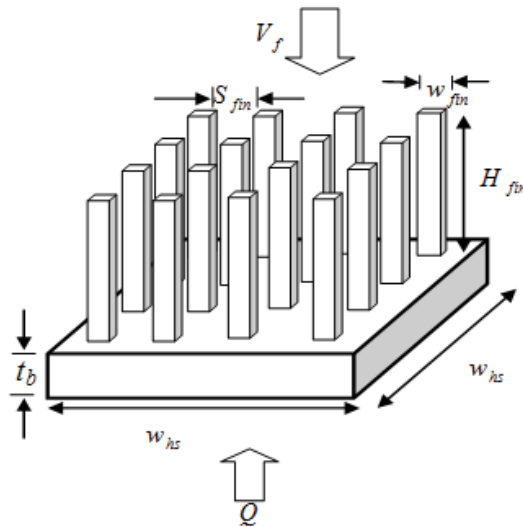


Fig. 4 Physical model of pin-fin heat sink

In the design process, an automated procedure was achieved by interfacing the CFD software into MATLAB. The design variables included those parameters for creating PFHS geometry and the inlet air velocity. For more details of the encoding/decoding process of the design variables, see [5].

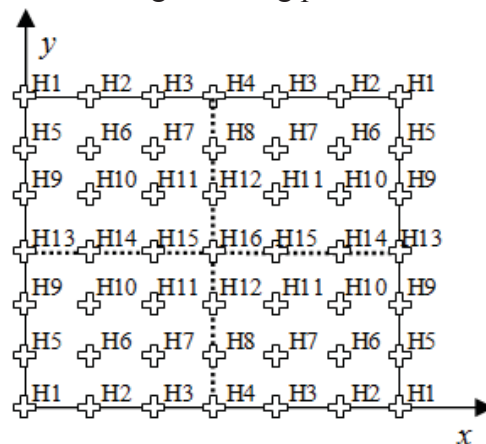


Fig. 5 Fin height distribution control points

The design problem was proposed to minimize bi-objective functions as fan pumping power and junction temperature of a pin-fin heat sink. The fan pumping power was used to measure the consumption of electronic power while the junction temperature determined heat sink thermal performance. Heat sink aspect ratio and space limit of the device were set to be design constraints. A mathematical model of the design problem can be expressed as follows:

Objective functions:

$$\min_{\mathbf{x}} \{T_j, P_{fan}\}$$

$$T_j = T_a + QR_{hs}$$

$$P_{fan} = \frac{\dot{m}_a \Delta P}{\rho_a}$$

Design variable vector:

$$\mathbf{x} = [n_f, w_{fin}, w_{hs}, V_f, H_{1-16}, t_b]^T$$

Constraints:

$$T_j - 355 \text{ °K} \leq 0$$

$$0.0025 \leq w_{fin} \leq 0.005$$

$$4 \leq N_f \leq 14$$

$$0.0025 \leq t_b \leq 0.005$$

$$0.0 \leq H_{fin} \leq 0.05$$

$$0.5 \leq V_f \leq 1.0$$

$$0.03 \leq w_{hs} \leq 0.06$$

$$0.0025 < s_{fin} < 0.004$$

In this work, 50 sampling points were created using a LHS technique while PBIL with 100 population size and 200 generations was performed with surrogate function evaluation. Then, 25 design solutions from the surrogate optimization were taken as initial solutions for a common PBIL, which was used for further exploring the non-dominated front with 25 population size, and 9 generations. This means that the proposed surrogate-assisted PBIL took $50 + 25 \times 10 = 300$ actual function evaluations.

Results and Discussion

In order to verify the proposed design approach, we also performed PBIL with the population size of 25 solutions and 35 generations resulting in 875 actual function evaluations. This was partly to ensure that the obtained non-dominated front was close to the real Pareto front. The front obtained from this run was termed PBIL-35 while the front obtained from using the surrogate-assisted PBIL was termed PBIL-KRG-10, and they are plotted in Fig. 6. From the figure, both fronts overlap each other which implies that one front was not absolutely better than the other. The hypervolume of the PBIL-35 front was 0.48699 while that of the PBIL-KRG-10 front was 0.46845 (the reference point was $f_1 = 303, f_2 = 0.14$). This implies that PBIL-35 was slightly better according to the hypervolume indicator.

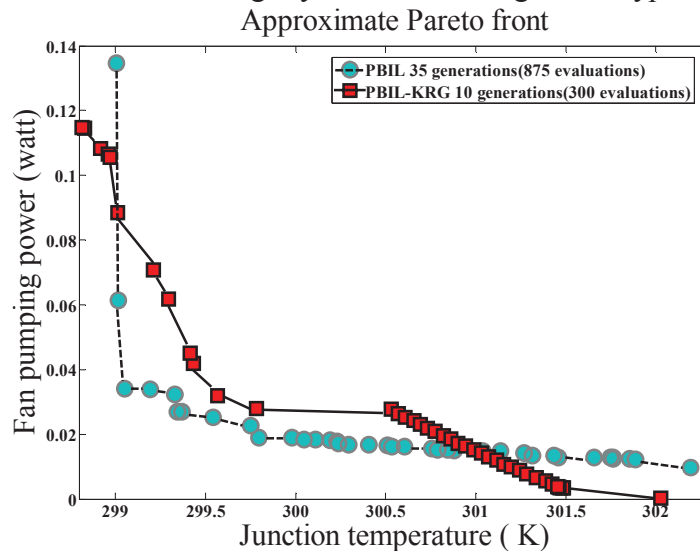


Fig. 6 Comparison of approximate Pareto fronts

Fig. 7 displays some selected heat sink geometries from the PBIL-KRG-10 front. It can be seen that the front gave various geometries with fin-height variations. Selected solutions from the PBIL-35 front are given in Fig. 8, which also have various fin height distributions. From both figures, it can be seen that heat sinks obtained from the surrogate-assisted approach tended to have more pin fins. However, each PFHS from both design approaches had fin height variation.

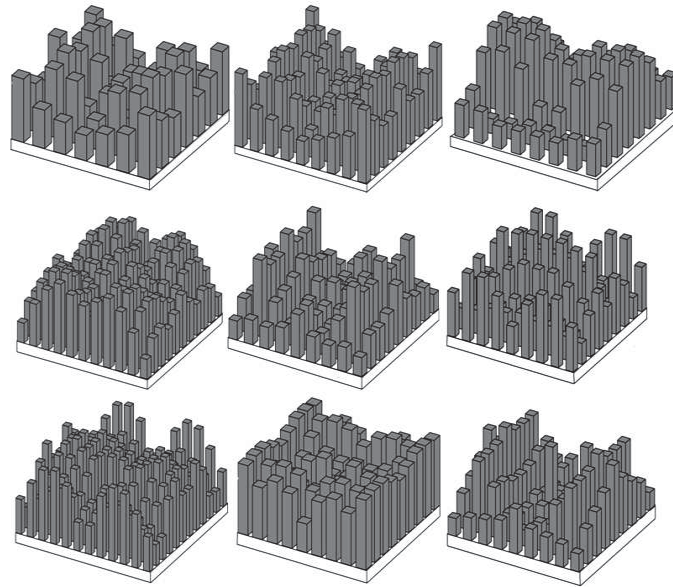


Fig. 7 Some optimal geometries from PBIL-KRG 10

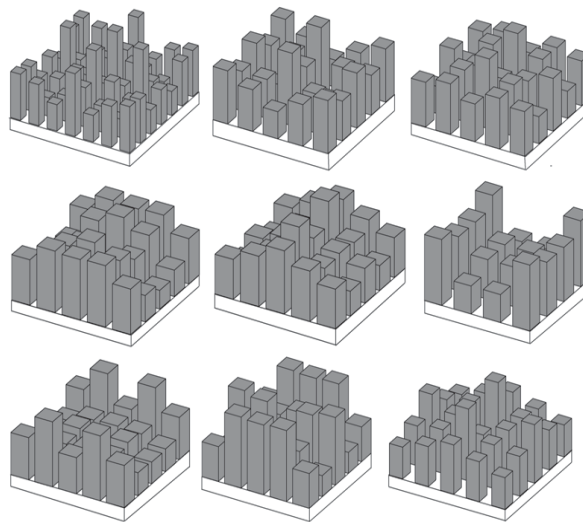


Fig. 8 Some optimal geometries from PBIL-35

Conclusions

The previous work shows that using solely PBIL to optimize a pin-fin heat sink is computationally expensive. Thus, the surrogate KRG model can be used for hybridization with PBIL to enhance its search performance. From the optimum results, it can be concluded that the surrogate-assisted PBIL is a powerful design strategy as the front obtained from this approach is comparable to that obtained from using the common PBIL with a considerable larger number of actual function evaluations. The former needs to perform three times less actual function evaluations approximately than the latter.

Acknowledgement

This research was supported by the Office of the Higher Education Commission, Thailand under the program Strategic Scholarships for Frontier Research Network (CHE-PhD-THA), the Sustainable Infrastructure Research and Development Center (Khon Kaen University), and the Thailand Research Fund (TRF).

References

- [1] S. Bureerat and S. Srisomporn: Engineering optimization Vol. 42 (2010), p. 305-323
- [2] K.T. Chiang: International communications in of heat and mass transfer Vol. 32 (2005), p. 1193-1201
- [3] W.A. Khan and M.M. Yovanovich: Journal of electronic packaging Vol. 130 (2008), p. 031010-1-031010-7
- [4] S. Srisomporn and S. Bureerat: IEEE Transaction on components and Packaging Technology Vol. 31 (2008), p. 351-360
- [5] S. Kanyakam and S. Bureerat: Inverse Problems in Design and Optimization Symposium, Brazil (2010)
- [6] Lophaven SN, Neilson HB, Sondergaard J. DACE a MATLAB Kriging Toolbox. Technical report IMM-TR-2002-12, Technical University of Denmark
- [7] Baluja, S.: Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. CMU_CS_163, (1994)
- [8] S. Bureerat and K. Sriworammas: Advances in Soft Computing Vol. 39(2007), pp. 223-232

Surrogate-Assisted Evolutionary Optimizers for Multiobjective Design of a Torque Arm Structure

Nantiwat Pholdee^a and Sujin Bureerat^b

Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand.

^anantiwat.ph@gmail.com, ^bsujbur@kku.ac.th

Keywords: Multiobjective population-based incremental learning, Latin Hypercube Sampling, Surrogate Model, Structural Constrained Optimization, Multiobjective Evolutionary Algorithms.

Abstract. This paper presents two surrogate-assisted optimization strategies for structural constrained multiobjective optimization. The optimization strategies are based on hybridization of multiobjective population-based incremental learning (MOPBIL) and radial-basis function (RBF) interpolation. The first strategy uses MOPBIL for generating training points while the second strategy uses a Latin hypercube sampling (LHS) technique. The design case study is the shape and sizing design of a torque arm structure. A design problem is set to minimize structural mass and displacement while constraints include stresses due to three different load cases. Structural analysis is carried out by means of a finite element approach. The design problem is then tackled by the proposed surrogate-assisted design strategies. Numerical results show that the use of MOPBIL for generating training points is superior to the use of LHS based on a hypervolume indicator and root mean square error (RMSE).

Introduction

In practice, an engineering design problem often has more than one design objective which is called multiobjective optimization. Such a design problem is assigned to optimize many objective functions simultaneously while fulfilling constraints. Among various types of optimizers, multiobjective evolutionary algorithms (MOEAs) are arguably the most popular method used to solve multiobjective problems. The most outstanding feature of MOEAs is that they can explore a Pareto front within a single run without requiring function derivatives. However, a lack of search consistency and a low convergence rate are the inevitable drawbacks. For this reason, the hybridization of a surrogate model (SM) and multiobjective evolutionary optimizers has been invented and this approach is found to be very powerful and effective [1].

A surrogate-assisted multiobjective evolutionary algorithm is widely used for design/optimization problems with expensive function evaluations or problems that have limited values of constraints and objective functions available. This kind of design process has been implemented on a wide variety of real world applications e.g. in [2-6]. The search performance of this approach relies on the accuracy of a surrogate model, which in turn depends on several factors such as the positions of the training points and the number of training points used. This paper compares two alternative techniques for generating sampling points for a surrogate-assisted approach: LHS and MOPBIL. RBF interpolation is the implemented surrogate model (SM). The two design strategies are implemented on a structural constrained optimization. The Pareto optimum results obtained by using the two strategies are compared and discussed.

Surrogate-Assisted MOPBIL

MOPBIL is chosen for this study as it is one of the most powerful MOEAs for solving a number of engineering applications [6-9]. A PBIL algorithm based on binary spaces is an evolutionary optimizer proposed as an alternative to a genetic algorithm. The concept of a PBIL is to use the so-called probability vector to represent a binary population. In multiobjective optimization, more probability

vectors should be used in order to obtain a better population diversity; thus, it is called a probability matrix [6, 9]. Starting with an initial probability matrix that has all elements as “0.5”, and an (empty) Pareto archive, the binary population according to the initial probability matrix is then created. The binary population is decoded and objective values are evaluated. The Pareto archive is then filled with non-dominated solutions obtained by sorting the combination of the members in the archive from the previous generation and the individuals in the current population. In cases where the total number of non-dominated solutions is greater than the predefined archive size, the archiving operator is used to remove some solutions from the archive. The Pareto archive is updated iteratively until reaching the maximum number of generations. More details of MOPBIL are given in [6, 9].

In this paper, the hybridization of a MOPBIL and a RBF interpolation model is proposed. Two simple hybrid strategies are as follows:

Strategy1. MOPBIL is used to explore sampling points for a few iterations. The obtained training points are then used to construct a RBF surrogate model. Afterwards, MOPBIL is used to find a non-dominated front of the design problem based on the surrogate model evaluations. Some solutions are selected from the members of the final archive where the actual function values of those solutions are determined. These selected solutions will be used as an initial non-dominated front for MOPBIL which will then proceed to further improve the front for a few iterations.

Strategy2. The second surrogate-assisted strategy is similar to Strategy1 except that Strategy2 uses LHS, one of the most commonly used techniques, for generating sampling points.

Design Problem

Constrained optimization of a torque arm structure, shown in Fig. 1, is chosen for the design demonstration and performance test. Under working conditions, the structure is subjected to three load cases as bending, torsion and axially compressive loads at its right hand cylindrical part.

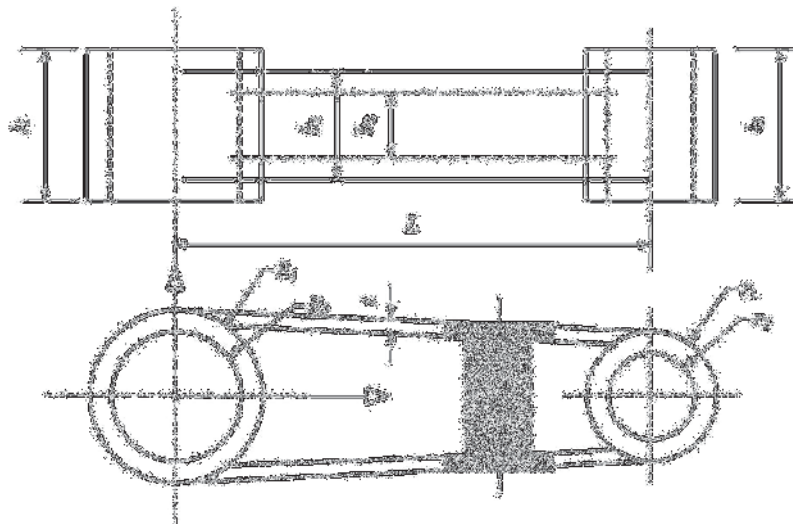


Fig.1 Geometry and parameterization of the torque arm.

The multiobjective optimization problem is posed to find the structural shape and size such that the mass and displacement are minimized whereas the constraint of stress failure is included which can be expressed as

$$\text{Min: } \mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x})] \quad (1)$$

Subject to

$$\sigma_{\max} \leq \sigma_{\text{all}}, \quad 0.03 \leq R_1 \leq 0.05, \quad 0.02 \leq r_1 \leq 0.05, \quad 0.01 \leq t \leq 0.025, \quad 0.035 \leq b_1 \leq 0.075, \\ 0.015 \leq b_2 \leq 0.03, \quad 0.035 \leq l_1 \leq 0.075, \quad 0.035 \leq l_2 \leq 0.075$$

where $\mathbf{x} = [R_1, r_1, t, b_1, b_2, l_1, l_2]^T$ is a design variable vector. SI units are used throughout unless otherwise specified. All the variables are displayed in Fig. 1. The variables R_2 and r_2 are set to be 0.0225 m and 0.015 m respectively. f_1 is the structural mass while f_2 is the maximum displacement due to the three load cases. Minimizing f_2 is somewhat equivalent to maximizing structural stiffness. The parameters $\sigma_{\max}$ and σ_{all} are the maximum von Mises stresses on the structure due to the three load cases and the allowable stress respectively.

The objective and constraint function values are evaluated by using finite element analysis (FEA). Function evaluation is somewhat time-consuming especially when the value of b_1 is close to l_1 or l_2 (but not equal). Strategy1 exploits MOPBIL for 3 iterations with a population size of 30 to find the sampling points. After performing MOPBIL in the surrogate environment, where a large number of function evaluations can be performed with much less computing time, 50 non-dominated solutions are selected where their actual function values are evaluated by means of FEA. After that, MOPBIL is operated for 5 iterations with a population size of 30 to further improve the non-dominated front. This means that Strategy1 needs a total of $30 \times 3 + 50 + 30 \times 5 = 290$ FEA evaluations. For Strategy2, the LHS technique is used to generate 100 sampling points instead of using MOPBIL for 3 iterations. The rest of the procedure is similar to Strategy1. Thus, Strategy2 requires 300 actual function evaluations. We also use MOPBIL without using a surrogate model to solve the problem (1) with the population size of 30 and 50 generations (a total of 1500 function evaluations). The obtained results can be used to measure the performance of the surrogate-assisted approaches.

Computational results

For the two surrogate-assisted design strategies, after an optimization process using a surrogate function prediction, 50 design solutions for each strategy are obtained and the actual function values are evaluated. The root mean square errors between the actual function values and the surrogate predicted values for the objective and constraint functions of both strategies are compared in Table 1. It can be seen that Strategy1 gives a better prediction for all functions except for the case of f_1 design objective. This implies that the MOPBIL generates a better set of sampling points.

Table 1. RMSE of constraint and objective functions: g_i = stress constraint due to the i -th load case

Methods	f_1	f_2	g_1	g_2	g_3
Strategy1	0.5193	0.6733×10^{-4}	6.6089×10^7	4.9847×10^7	5.4952×10^7
Strategy2	0.4902	0.7026×10^{-4}	8.4202×10^7	8.4089×10^7	9.3479×10^7

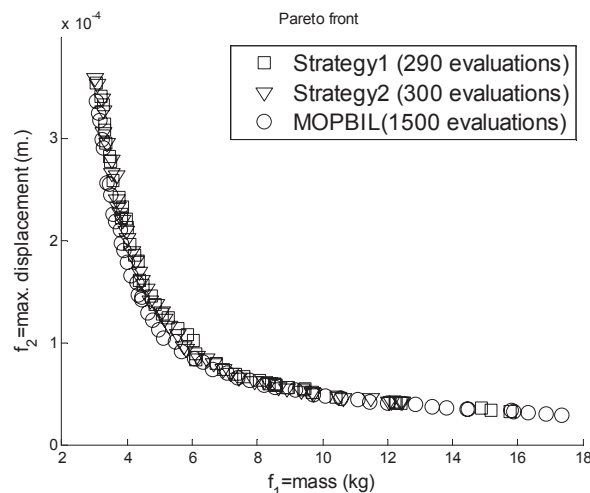


Fig.2 Pareto optimal front obtained.

The final Pareto optimal fronts of Strategy1, Strategy2, and MOPBIL are given in Fig. 2. It can be seen that all three fronts give similar results. The hypervolumes of the fronts obtained by using Strategy1, Strategy2, and MOPBIL are 0.0027, 0.0024, and 0.0025 respectively given that the reference point is $[f_1 = 17 \text{ kg}, f_2 = 0.00025 \text{ m}]$. As a result, the front from Strategy1 is the best according to the hypervolume indicator.

Figs. 3-5 respectively show the variation of the design variables in the non-dominated fronts obtained from Strategy1, Strategy2, and MOPBIL. The ranges of the design variables for the cases of Strategy1 and Strategy2 are very similar while the ranges of the variables for the case of MOPBIL are slightly different but still have the same trends.

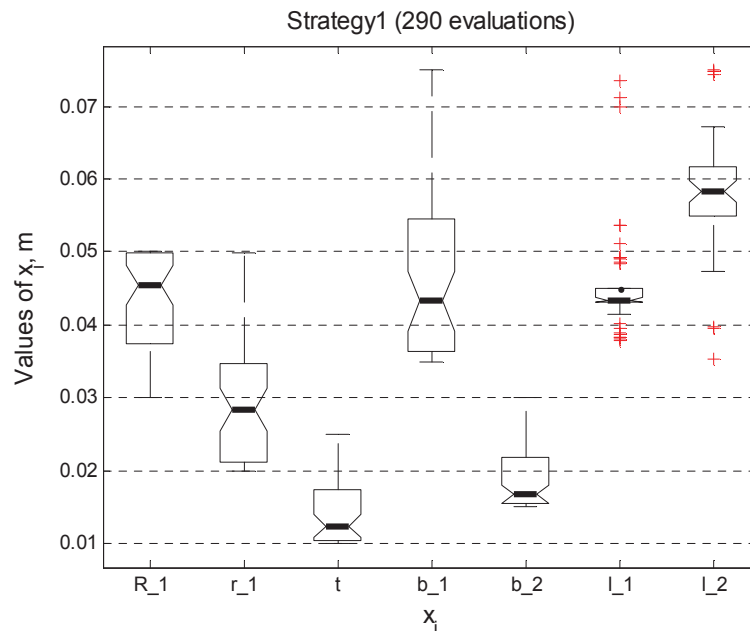


Fig. 3 Variation of design variables in the front from Strategy1

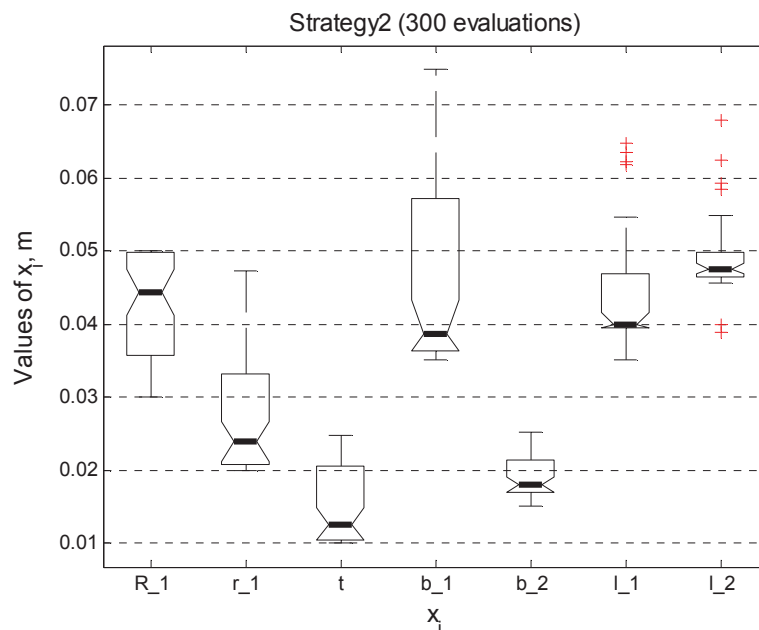


Fig. 4 Variation of design variables in the front from Strategy2

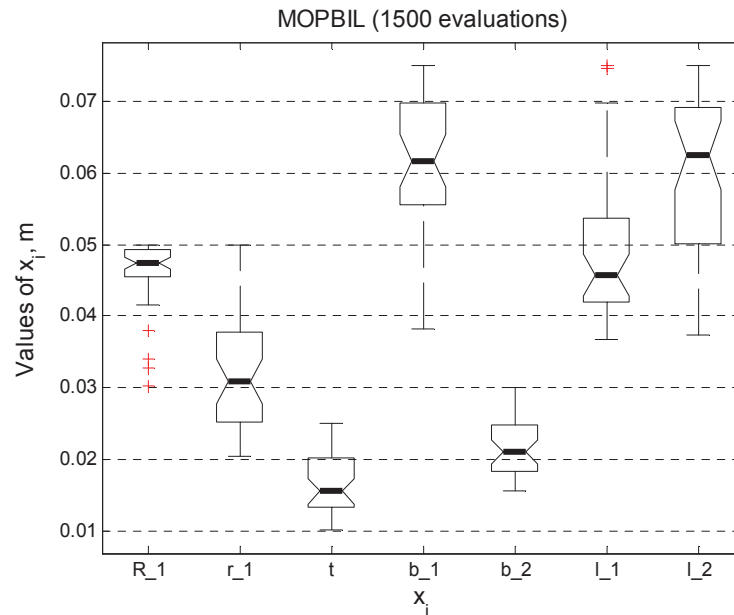


Fig. 5 Variation of design variables in the front from MOPBIL

Conclusions

The multiobjective shape and sizing optimization problem of a torque arm using the hybridization of a RBF surrogate model and MOPBIL is demonstrated. From the comparative results, the surrogate-assisted approach using MOPBIL for generating training points is slightly better than the one using the LHS technique. The results obtained from using both surrogate-assisted design strategies are comparable to those obtained from using MOPBIL without a surrogate hybrid although the formers employ a lower number of function evaluations. The proposed surrogate-assisted MOPBIL strategies are said to be efficient but, from RMSE, the surrogate prediction still needs further improvement especially the stress constraint functions.

Acknowledgments

The authors are grateful for the support from The Royal Golden Jubilee PhD Program (RGJ-TRF) and the Thailand Research Fund (TRF).

References

- [1] S. Srisomporn and S. Bureerat: IEEE TRANSACTIONS ON COMPONENTS AND PACKAGING TECHNOLOGIES Vol. 31(2008), pp. 351-360
- [2] P. Breitkopf, H. Naceur, A. Rassineux and P. Villon: Computers and Structures Vol. 83 (2005), pp. 1411-1428.
- [3] K. T. Chiang and F. P. Chang: International Communications in Heat and Mass Transfer, Vol. 33 (2006), pp. 836-845.
- [4] H. Naceur, S. Ben-Elechi, J. L. Batoz and C. Knopf-Lenoir: Materials and Design Vol. 29 (2008), pp. 781-790
- [5] W. Hu, L. G. Yao, and Z. Z Hua: Journal of materials processing technology Vol. 197(2008), pp. 77-88
- [6] S. Bureerat and S. Srisomporn: Engineering Optimization Vol. 42(2010), pp. 305-343
- [7] S. Kanyakam and S. Bureerat: submitted to 2007 IEEE Congress on Evolutionary Computation, Singapore, 25-28 September 2007, pp. 4162 - 4169
- [8] C.Noilublao and S.Bureerat: Evolutionary Computation In-Tech (2009), pp. 487-580
- [9] S. Bureerat and K. Sriworamas: Advances in Soft Computing Vol. 39(2007), pp. 223-232

Advances in Engineering Design and Optimization II

doi:10.4028/www.scientific.net/AMM.101-102

Surrogate-Assisted Evolutionary Optimizers for Multiobjective Design of a Torque Arm Structure

doi:10.4028/www.scientific.net/AMM.101-102.324

Hybrid Population-Based Incremental Learning Using Real Codes

Sujin Bureerat

Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University,
40002, Thailand
sujbur@kku.ac.th

Abstract. This paper proposes a hybrid evolutionary algorithm (EA) dealing with population-based incremental learning (PBIL) and some efficient local search strategies. A simple PBIL using real codes is developed. The evolutionary direction and approximate gradient operators are integrated to the main procedure of PBIL. The method is proposed for single objective global optimization. The search performance of the developed hybrid algorithm for box-constrained optimization is compared with a number of well-established and newly developed evolutionary algorithms and meta-heuristics. It is found that, with the given optimization settings, the proposed hybrid optimizer outperforms the other EAs. The new derivative-free algorithm can maintain outstanding abilities of EAs.

Keywords: Population-Based Incremental Learning, Approximate Gradient, Evolutionary Direction, Meta-Heuristics, Evolutionary Algorithms.

1 Introduction

Evolutionary algorithms (EAs) or meta-heuristic search algorithms are commonly known as alternative optimizers to classical mathematical programming (MP) or gradient-based optimizers. Using EAs is advantageous over MP since they are simple to implement, more robust, capable of tackling global optimization, and derivative-free. Nevertheless, the methods have some unavoidable disadvantages since they have a low convergence rate and require a large number of function evaluations to achieve optimum results. With no guarantee of convergence, the optimum results obtained from using EAs are usually classified as near optima. EAs also have a complete lack of search consistency since, with multiple simulation runs, they are unlikely to find the same optimum point. As a result of the attractiveness of their ability to tackle almost all kinds of optimization problems and the aforementioned advantages, many researchers and engineers have invested considerable effort to improve and develop evolutionary optimizers. The target is to retain their outstanding abilities and alleviate their drawbacks. From genetic algorithms (GA) to the countless number of EAs presently being used in a wide variety of real world applications, only those EAs with a high searching performance are receiving considerable attention.

Several methods have been used to enhance the search performance of EAs. One of the most popular and efficient strategies is the use of EAs in combination with

surrogate models [1, 2]. This hybridization approach is said to be well-established and successfully implemented on a variety of real world applications. Another approach is the integration between EAs and their variants. Since the weak and strong points of EA operators have been thoroughly investigated, their proper combinations can also improve EA performance [3].

This paper proposes a hybrid evolutionary algorithm dealing with population-based incremental learning and some efficient local search strategies. The method is developed to deal with single objective global optimization. A simple PBIL using real codes is detailed. An evolutionary direction operator and an approximate gradient are integrated with the main procedure of PBIL. The search performance of the developed hybrid algorithm while solving 35 box-constrained optimization problems is compared with a number of well-established and newly developed EAs and meta-heuristics. It is found that the proposed optimizer can be regarded as one of the best evolutionary optimizers.

The paper is organized as follows. The following section 2 gives the details of population-based incremental learning using real codes, evolutionary direction and approximate gradient operators, and the hybrid algorithm. Section 3 provides the testing functions for performance comparison. Section 4 shows the comparative results and assessments of the EAs performance. The paper is concluded in section 5.

2 Hybrid Algorithm

2.1 Population-Based Incremental Learning

The population-based incremental learning was first developed by Baluja as an alternative search algorithm to genetic algorithm [4]. Unlike most traditional EAs, PBIL uses the so-called probability vector to estimate a binary population. The method accomplishes optimization search by improving the probability vector iteratively. The real-code variants of PBIL have been developed [5-6] but they seem to be less popular than the original binary-code PBIL. In this paper, we propose PBIL using real code, which exploits the probability matrix similar to the histogram PBIL (PBIL_H) in [6]. Given that the box-constrained optimization problem is of the form:

$$\begin{aligned} \text{Min } f(\mathbf{x}) \\ L_i \leq x_i \leq U_i; i = 1, \dots, n \\ \mathbf{x} \in R^n \end{aligned} \quad (1)$$

where $\mathbf{x}$ is the vector of design variables size $n \times 1$, f is an objective function, L_i are the lower bounds of $\mathbf{x}$, and U_i are the upper bounds of $\mathbf{x}$.

The probability matrix ($\mathbf{P}$ or P_{ij} size $n \times m$) is proposed to deal with real design parameters in such a way that the feasible range $[L_i, U_i]$ of a design variable is divided into m sections. The element P_{ij} determines the probability that the i -th element of $\mathbf{x}$ will be placed in the range $[L_i + (j-1)\delta_i, L_i + j\delta_i]$ where $\delta_i = (U_i - L_i)/m$. Generation of a real-code population can be carried out in a similar manner as with binary PBIL. A real-code PBIL search starts with a probability matrix $\mathbf{P}$ where all elements values are assigned as $1/m$. An initial population according to $\mathbf{P}$ is then created with their corresponding objectives being evaluated while the best individual $\mathbf{x}^{best}$ is detected. The probability matrix is then updated based upon $\mathbf{x}^{best}$ as

$$P'_{ij} = (1 - L_{R,j})P_{ij}^{old} + L_{R,j} \quad (2)$$

where

$$L_{R,j} = 0.5 \exp(-(j - r)^2). \quad (3)$$

The element r is determined in such a way that x_i^{best} is placed in the range $[L_i + (r-1)\delta_i, L_i + r\delta_i]$. The learning rate L_R in Eq. 3 is set to prevent premature convergence.

In order to preserve the condition $\sum_{j=1}^T P_{ij} = 1$, the i -th row of $\mathbf{P}'$ is normalized as:

$$P''_{ij} = (1 / \sum_{j=1}^m P'_{ij}) P'_{ij} \quad (4)$$

As a result, the finally updated P_{ij} is in Eq. 4. The probability matrix and $\mathbf{x}^{best}$ are iteratively improved until the termination condition is met. The pseudo-code of real-code PBIL is given in Fig. 1 where t is a generation number, N_G is the total number of generations, and N_P is the population size.

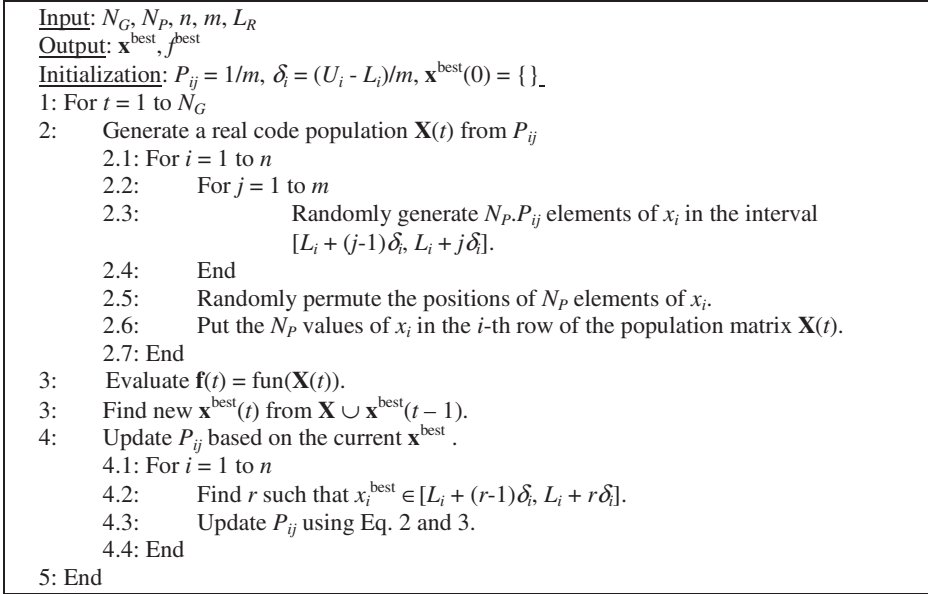


Fig. 1. Algorithm for real code population-based incremental learning

2.2 Evolutionary Direction Recombination

The evolutionary direction operator was proposed in [7], which is the modification of the work in [8]. It can be thought of as a special kind of evolutionary recombination. One operation requires three randomly selected individuals from the current

population to produce a pair of children. Let the three individuals be $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ where $\mathbf{x}_1$ has the best (minimum) objective among them. An evolutionary direction is computed as:

$$\mathbf{s} = (\mathbf{x}_1 - \mathbf{x}_2) + (\mathbf{x}_1 - \mathbf{x}_3) + \mathbf{c} \quad (5)$$

where $c_i = \varepsilon \cdot \text{randn}$, randn is a normally distributed random number with mean zero and standard deviation one, and ε is a small number to be specified (default value is 0.05). The random vector $\mathbf{c}$ is used to prevent a premature convergence. The new individuals as the product of this evolutionary operator can be obtained as

$$\mathbf{y}_1 = \mathbf{x}_1 + \text{rand} \cdot \lambda(t) \mathbf{s} \quad (6)$$

$$\mathbf{y}_2 = \mathbf{x}_1 - \text{rand} \cdot \lambda(t) \mathbf{s}$$

where $\text{rand} \in [0,1]$ is a uniform random number, and $\lambda(t)$ is the maximum step length at the t -th generation. The value of λ is set to have greater value earlier. As the optimization progresses, it becomes smaller. In this work, the maximum step length is set to be

$$\lambda(t) = \exp\left(\frac{6.2146}{N_G - 1} - 0.6931\right) \exp\left(-\frac{6.2146}{N_G - 1} t\right), \quad (7)$$

which means $\lambda(1) = 0.5$ and $\lambda(N_G) = 0.001$.

This strategy is set for local search. Note that the solutions will be treated to satisfy the bound constraints before performing function evaluation. Fig. 2 illustrates the evolutionary operation given that $\mathbf{c}$ in Eq. 5 is set to be a zero vector for simplicity. In Fig. 2 a) and b), the solid line arrow is the search direction for $\mathbf{y}_1$ whereas the dashed arrow is the search direction for $\mathbf{y}_2$. In Fig. 2 a), it is shown that the offspring $\mathbf{y}_1$ is better than $\mathbf{y}_2$. However, in Fig. 2 b), $\mathbf{y}_2$ has the possibility to be better than $\mathbf{y}_1$ and for this reason the + and - signs are used in Eq. 6.

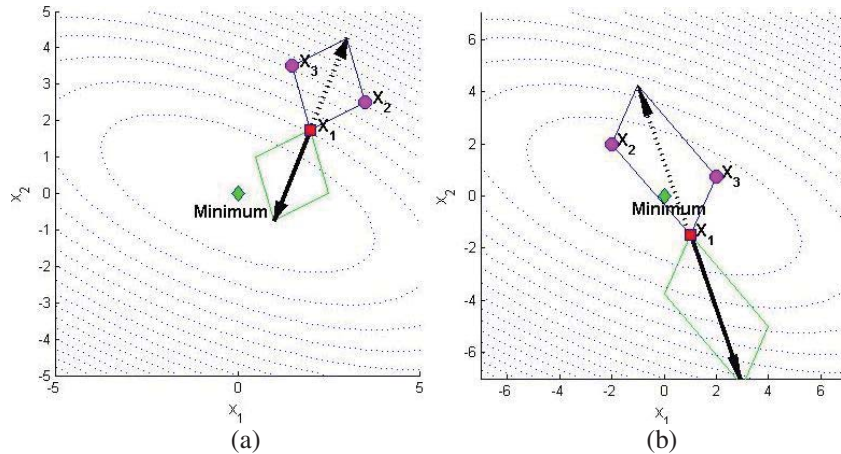


Fig. 2. Evolutionary directions

2.3 Approximate Gradient

The approximate gradient is estimated from the members of a population and their corresponding objectives [7]. The approximate gradient is calculated exploiting the relation of the directional derivative. The directional derivative of a function f in the direction of $\mathbf{s}$ at a point $\mathbf{x}$ in R^n space, denoted by df/ds can be expressed as:

$$\mathbf{u}_s^T \nabla f = \frac{df}{ds} \approx \frac{f(\mathbf{x}) - f(\mathbf{x} + \Delta \mathbf{x})}{\Delta s} \quad (8)$$

where $\mathbf{u}_s$ is a unit vector of $\mathbf{s}$. With the current population size $n \times N_p$ $\{\mathbf{x}_1, \dots, \mathbf{x}_{NG}\}$, their objectives $\{f_1, \dots, f_{NG}\}$, the current best solution $\mathbf{x}^{best}$, and its objective f^{best} , the approximate gradient of f at the point $\mathbf{x}^{best}$ can be computed as:

$$\begin{bmatrix} (\mathbf{x}^{best} - \mathbf{x}_1)^T \\ \vdots \\ (\mathbf{x}^{best} - \mathbf{x}_R)^T \end{bmatrix} \nabla f = \mathbf{A} \nabla f = \begin{bmatrix} f^{best} - f_1 \\ \vdots \\ f^{best} - f_R \end{bmatrix} = \mathbf{b}. \quad (9)$$

The first R individuals that are closest to $\mathbf{x}^{best}$ are chosen to approximate the gradient. The value R is set to be greater than n in order to prevent matrix singularity. Since the matrix $\mathbf{A}$ is not a square matrix, Eq. 9 can be solved using the pseudo-inverse operation; thus, the approximate gradient is termed pseudo-gradient in [7]. Note that pseudo-inverse algorithms are available in both free and commercial software such as SCILAB and MATLAB. Since ∇f is used for local search rather than global search, it is also useful to apply a quadratic interpolation to enhance the search efficiency. Based on the steepest descent method, the search direction is set to be $\mathbf{s} = -\nabla f$. Two solutions extended from $\mathbf{x}^{best}$ along the search direction are found as follows:

$$\mathbf{z}_1 = \mathbf{x}^{best} + \beta_1 \cdot \mathbf{s} \quad (10)$$

$$\mathbf{z}_2 = \mathbf{x}^{best} + \beta_2 \cdot \mathbf{s}$$

where $\beta_1 \neq \beta_2$ are two randomly generated numbers in the range of (0,1]. The third solution for this process can be determined by applying a quadratic interpolation technique. We can assume that the objective function along the search direction is a quadratic function of the variable β as

$$f(\mathbf{x}^{best} + \beta \cdot \mathbf{s}) = C_1 \beta^2 + C_2 \beta + C_3. \quad (11)$$

The quadratic function coefficients can be found by solving

$$\begin{bmatrix} 0 & 0 & 1 \\ \beta_1^2 & \beta_1 & 1 \\ \beta_2^2 & \beta_2 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} f^{best} \\ f(\mathbf{z}_1) \\ f(\mathbf{z}_2) \end{bmatrix} \quad (12)$$

where $f(\mathbf{z}_1)$ and $f(\mathbf{z}_2)$ are the objective function values at the points $\mathbf{z}_1$ and $\mathbf{z}_2$ respectively. Then, the third individual can be found

$$\mathbf{z}_3 = \mathbf{x}^{best} + \beta_3 \mathbf{s} \quad (13)$$

where $\beta_3 = -C_2/2/C_1$.

This is equivalent to performing the Powell line search method for one step. In cases that $C_1 = 0$, which implies that the objective could be a linear function, t_3 is generated at random. Fig. 3 shows the process of evaluating $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ of the approximate gradient operator. The solution $\mathbf{z}_3$ is treated to be inside the bounds before performing function evaluation. For $\mathbf{z}_1$ and $\mathbf{z}_2$, if they are located outside the feasible region, they will be discarded from the optimization process. Nevertheless, their function evaluations are counted to the total number of function evaluations.

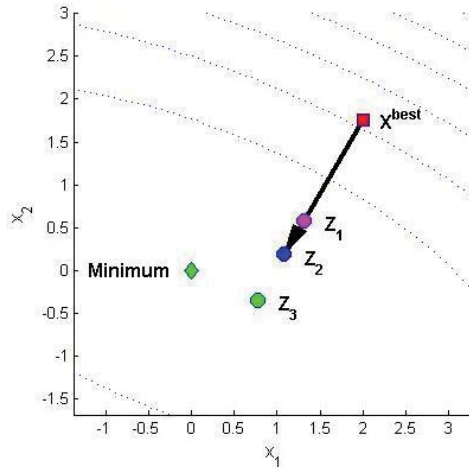


Fig. 3. Step length determination of an approximate gradient operator

2.4 Hybrid Algorithm

The algorithm of the hybrid PBIL is given in Fig. 4. Initially, the hybrid algorithm starts with an initial probability matrix of PBIL, an initial real-code population, and an initial best design solution $\mathbf{x}^{best}$. $N_p/2$ individuals as the first sub-population are then created according to the probability matrix. An approximate gradient is estimated while three new solutions are created from this operation. The rest of the solutions (approximately $N_p/2 - 3$ solutions) to fill in the current population are created by using the evolutionary direction operator. After combining the three sub-populations, the function evaluation is performed. Afterwards, the best individual is detected and used to update the probability matrix and compute an approximate gradient direction. The process is repeated until the maximum number of iterations is reached. In this hybrid algorithm, three sub-populations are created in parallel but the operators share information during the search since updating a probability matrix and computing an approximate gradient require $\mathbf{x}^{best}$ for the operation.

Input: $N_G, N_P, n, m, L_R, \varepsilon$
Output: $\mathbf{x}^{\text{best}}, f^{\text{best}}$
Initialization: $P_{ij} = 1/m, \delta_i = (U_i - L_i)/m, \mathbf{X}(0), \mathbf{f}(0), \mathbf{x}^{\text{best}}(0)$
1: For $t = 1$ to N_G
2: Generate a sub-population $\mathbf{X}_1$ ($N_P/2$ solutions) from P_{ij} following the computational steps 2.1-2.7 in Fig. 1.
3: Compute an approximate gradient using Eq. 9.
4: Generate 3 individuals $\mathbf{X}_2$ from an approximate gradient operator.
5: Generate $N_P/2 - 3$ individuals $\mathbf{X}_3$ from an evolutionary direction operator.
6: Combine $\mathbf{X}(t) = \mathbf{X}_1 \cup \mathbf{X}_2 \cup \mathbf{X}_3$, and evaluate $\mathbf{f}(t) = \text{fun}(\mathbf{X}(t))$.
7: Find new $\mathbf{x}^{\text{best}}(t)$ from $\mathbf{X} \cup \mathbf{x}^{\text{best}}(t-1)$.
8: Update P_{ij} based on the current $\mathbf{x}^{\text{best}}$ using the computational steps 4.1 - 4.4 in Fig. 1.
9: End

Fig. 4. Algorithm of the Hybrid PBIL

3 Testing Functions

In order to examine the searching performance of the proposed hybrid algorithm, 35 testing functions of box-constrained optimization are posed as detailed in Table 1. With the exception of F_8 [9], all of the functions are taken from [10] and [11] where their expressions and more details can be tracked back from their given names. Evolutionary optimizers used to compare with the hybrid algorithm are as follows:

Real-code ant colony optimization (ACO) [10]: The parameters used for computing the weighting factor and the standard deviation in the algorithm are set to be $\xi = 1.0$ and $q = 0.2$ respectively.

Charged system search (CSS) [12]: The number of solutions in the charge memory is $0.2N_P$. The charged moving considering rate and the parameter PAR are set to be 0.75 and 0.5 respectively.

Differential evolution (DE) [13]: DE step size, crossover probability, and refresh iterations are set as 0.8, 0.5, and 10 respectively. The DE/rand/1/bin strategy is used.

Continuous tabu search (TS) [14]: The sizes of tabu list and promising list are set to be $2N_P$ and $0.5N_P$ respectively.

Fireworks algorithm (FA) [15]: The number of fireworks for each generation is $0.25N_P$, the limit of sparks created with algorithm 1 is N_P , the amount of sparks created with algorithm 2 is $0.25N_P$, the floor parameter for rounding the amount of sparks created with algorithm 1 is 0.004, the ceiling parameter for rounding the amount of sparks created with algorithm 1 is 0.8, and the maximum explosion amplitude is $0.5(U_i - L_i)$.

Binary-code genetic algorithm (GA) [9]: The crossover and mutation probabilities are 1.0 and 0.1 respectively.

Particle swarm optimization [16]: The starting inertia weight, ending inertia weight, cognitive learning factor, and social learning factor are assigned as 0.5, 0.01, 0.5 and 0.5 respectively.

Simulated annealing (SA) [17]: During an optimization run, an annealing temperature is reduced exponentially 10 times from the value of 10 to 0.001. On each loop, $2n$ children are created by means of mutation to be compared with their parent.

Continuous scatter search (SS) [11]: The BLX- α recombination method is used. The number of high-quality solutions in the reference set is $0.25N_p$, and the number of diverse solutions in the reference set is $0.25N_p$.

Binary PBIL (PB) [4]: The learning rate, mutation shift, and mutation probability are set as 0.5, 0.2, and 0.05 respectively.

The two algorithms from this paper are real-code PBIL (PR) as detailed in subsection 2.1, and real code PBIL in combination with the evolutionary direction and approximate gradient operators (HPR). The number of columns of the probability matrix is set to be $10n$. Each method is used to solve each optimization problem 30 runs starting with the same initial population. The best results the methods can search for are taken as near optimum solutions. The number of iterations is set to be $10n$ whereas the population size is $7n$. In cases of EAs and meta-heuristics that use different search strategies such as fireworks algorithm, simulated annealing, continuous scatter search, and charged system search, the number of iterations and population size may not be the same values as previously mentioned but they will use the same total number of function evaluations i.e. $10n \times 7n$ evaluations.

Table 1. Testing functions

Function no., details	$[L,U]^n$	Function no., details	$[L,U]^n$
1, B_2 [10]	$[-50,100]^2$	18, Griewangk [10]	$[-5.12,5.12]^{10}$
2, Beale [11]	$[-4.5,4.5]^2$	19, <i>Perm</i> [11]	$[-15,15]^{15}$
3, Booth [11]	$[-10,10]^2$	20, <i>Perm0</i> [11]	$[-15,15]^{15}$
4, Easom [10]	$[-100,100]^2$	21, Cigar [10]	$[-3,7]^{20}$
5, Goldstein & Price [10]	$[-2,2]^2$	22, Diagonal plane [10]	$[0.5,1.5]^{20}$
6, Martin & Gaddy [10]	$[-20,20]^2$	23, Dixon & Price [11]	$[-10,10]^{20}$
7, Matyas [11]	$[-5,10]^2$	24, Levy(n) [11]	$[-10,10]^{20}$
8, Penny & Linfield* [9]	$[-5,5]^2$	25, Powell(n) [11]	$[-4,5]^{20}$
9, Powersum [11]	$[0,2]^2$	26, Rastrigin(n) [10]	$[-2.56,5.12]^{20}$
10, Branin [10]	$[-5,15]^2$	27, Rosenbrock [10]	$[-5,5]^{20}$
11, Shubert [11]	$[-10,10]^2$	28, Sum Squares [11]	$[-5,10]^{20}$
12, Six Hump Camel Back [11]	$[-5,10]^2$	29, Schwefel [11]	$[-500,500]^{20}$
13, Colville [11]	$[-10,10]^4$	30, Trid(n) [11]	$[-400,400]^{20}$
14, Hartmann 3,4 [10]	$[0,4]^4$	31, Ackley(n) [11]	$[-15,30]^{30}$
15, Shekel [10]	$[0,10]^4$	32, Ellipsoid [10]	$[-3,7]^{30}$
16, Zakharov [10]	$[-5,10]^5$	33, Plane [10]	$[-0.5,1.5]^{30}$
17, Hartmann 6,4 [10]	$[0,6]^6$	34, Sphere [10]	$[-3,7]^{30}$
		35, Tablet [10]	$[-3,7]^{30}$

* $f_8(\mathbf{x}) = 0.5(x_1^4 - 16x_1^2 + 5x_1) + 0.5(x_2^4 - 16x_2^2 + 5x_2)$

4 Comparison Results

For each testing function, each method will produce 30 near optimum values. The average value of 30 near optimum values obtained from using 12 EAs is found and normalized using the relation

Table 2. Comparative results by normalized function values

Fn No.	ACO	CSS	DE	TS	FA	GA	PSO	SA	SS	PB	PR	HPR
1	0.000	0.035	0.040	0.016	0.050	0.869	0.018	0.032	0.105	1.000	0.019	0.005
2	0.020	0.520	1.000	0.130	0.220	0.265	0.000	0.148	0.542	0.302	0.431	0.058
3	0.014	0.056	0.166	0.005	0.056	1.000	0.015	0.353	0.413	0.778	0.014	0.000
4	0.829	1.000	0.830	0.553	0.997	0.550	0.283	0.988	1.000	0.912	0.529	0.000
5	0.015	1.000	0.439	0.000	0.090	0.836	0.520	0.718	0.687	0.841	0.024	0.144
6	0.001	0.051	0.068	0.003	0.037	0.400	0.001	0.083	0.089	1.000	0.005	0.000
7	0.036	0.109	0.293	0.006	0.099	0.500	0.006	0.501	0.578	1.000	0.016	0.001
8	0.085	0.074	0.051	0.516	0.284	0.159	1.000	0.000	0.193	0.185	0.190	0.188
9	0.009	0.030	0.132	0.002	0.043	0.440	0.119	1.000	0.153	0.658	0.002	0.002
10	1.000	0.076	0.516	0.016	0.149	0.421	0.163	0.069	0.290	0.755	0.041	0.000
11	0.934	0.520	0.862	0.165	0.315	0.346	0.539	0.116	1.000	0.570	0.340	0.000
12	0.038	0.801	0.476	0.273	0.363	0.536	0.998	0.337	0.749	1.000	0.125	0.000
13	0.239	0.186	1.000	0.031	0.319	0.329	0.141	0.154	0.337	0.344	0.165	0.000
14	0.000	1.000	0.002	0.005	0.006	0.036	0.301	0.439	0.023	0.203	0.084	0.004
15	0.052	0.000	0.821	0.623	0.885	0.634	0.454	0.920	0.377	1.000	0.772	0.573
16	0.033	0.017	0.412	0.001	0.020	0.632	0.339	0.180	0.205	1.000	0.006	0.000
17	0.000	1.000	0.051	0.237	0.129	0.191	0.908	0.001	0.531	0.405	0.251	0.247
18	0.704	0.000	1.000	0.163	0.175	0.076	0.150	0.161	0.015	0.040	0.173	0.097
19	0.000	0.462	0.000	0.546	0.619	0.027	0.890	0.000	0.267	0.081	1.000	0.417
20	0.000	0.000	0.002	0.000	0.000	0.000	0.000	1.000	0.000	0.001	0.000	0.000
21	0.000	0.000	0.036	0.002	0.026	0.338	0.179	0.001	0.199	1.000	0.043	0.000
22	0.000	0.014	0.000	0.006	0.000	0.001	1.000	0.000	0.057	0.022	0.022	0.000
23	0.009	0.000	0.114	0.000	0.013	0.166	0.308	0.003	0.022	1.000	0.042	0.007
24	0.030	0.023	0.088	1.000	0.886	0.059	0.320	0.000	0.042	0.212	0.403	0.408
25	0.047	0.000	0.147	0.006	0.011	0.420	0.712	0.005	0.049	1.000	0.061	0.001
26	1.000	0.386	0.588	0.601	0.485	0.176	0.477	0.000	0.470	0.293	0.722	0.379
27	0.076	0.011	0.759	0.000	0.081	0.481	1.000	0.004	0.117	0.860	0.153	0.087
28	0.001	0.000	0.042	0.002	0.024	0.149	0.810	0.001	0.157	1.000	0.036	0.000
29	1.000	0.266	0.468	0.449	0.485	0.130	0.827	0.000	0.795	0.223	0.679	0.507
30	0.110	0.333	0.080	0.429	0.000	0.410	1.000	0.454	0.698	0.515	0.332	0.203
31	0.279	0.074	0.519	0.323	0.264	0.537	1.000	0.000	0.642	0.889	0.331	0.079
32	0.004	0.000	0.041	0.001	0.036	0.119	1.000	0.000	0.179	0.452	0.031	0.004
33	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
34	0.007	0.000	0.065	0.000	0.009	0.106	1.000	0.000	0.187	0.319	0.016	0.000
35	0.000	0.000	0.004	0.008	0.001	0.030	1.000	0.000	0.025	0.054	0.232	0.002
total	6.573	8.043	11.113	6.118	7.177	11.370	17.477	7.669	11.192	19.915	7.291	3.414

* PB = binary PBIL, PR = real code PBIL, HPR =Hybrid real-code PBIL

$$\bar{f}_i = \left| \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right| \quad (14)$$

where $f_{\min}$ is the average near optimum value of the best method, and $f_{\max}$ is the average near optimum value of the worst method. By using Eq. 12, the best method will have $\bar{f}=0$ whereas the worst method will have $\bar{f}=1$. This relative comparison is given in Table 2 where each value in the table stands for a $\bar{f}$ value. From the results, the overall top five best performers are the proposed hybrid PBIL algorithm,

continuous tabu search, real-code ant colony optimization, fireworks algorithm, real-code PBIL, and simulated annealing. Clearly, there is no absolute best method. The proposed hybrid approach is slightly ahead of the second best TS and the third best ACO. Charged system search and simulated annealing are the two best methods for large scale objective functions having one optimum. The continuous tabu search uses the longest computational time for each optimization run. The hybrid PBIL takes a slightly longer time than the real-code PBIL. Among the PBIL variants, real-code PBIL outperforms its binary-code counterpart. The searching performance of PBIL is improved when integrated with the evolutionary direction and approximate gradient operators. This ranking is made to show the convergence rate of EAs, which means the hybrid PBIL has a high convergence rate when compared to the other EAs with the given optimization parameters and conditions.

An alternative EA performance assessment is given in Table 3-4. Table 3 shows the ranking of the 12 implemented evolutionary algorithms for the first test function. Firstly the performance matrix size 12×12, whose elements are full of zeros, is generated. Then, the results obtained from method *I* and method *J* are compared using the statistical *t*-test at 95% confidence level. In cases that the mean objective function value from method *I* is significantly different from that obtained from method *J*, an element of the performance matrix is modified. The element at row *I* and column *J* of the matrix is changed to be one if the mean value obtained from method *I* is higher; otherwise, element at row *J* and column *I* is changed to be one. Having a complete performance matrix, the values on each column are summed up. The algorithm having that highest score (ACO in Table 3) is considered the best method while the method having the lowest total value is the worst for solving this test function.

Table 3. Performance matrix and ranking score using t-test: Function number 1

EAs	ACO	CSS	DE	TS	FA	GA	PSO	SA	SS	PB	PR	HPR
ACO	0	0	0	0	0	0	0	0	0	0	0	0
CSS	1	0	0	0	0	0	0	0	0	0	0	1
DE	1	0	0	1	0	0	1	0	0	0	1	1
TS	1	0	0	0	0	0	0	0	0	0	0	1
FA	1	0	0	1	0	0	1	0	0	0	1	1
GA	1	1	1	1	1	0	1	1	1	0	1	1
PSO	1	0	0	0	0	0	0	0	0	0	0	0
SA	1	0	0	1	0	0	0	0	0	0	0	1
SS	1	0	0	1	0	0	1	0	0	0	1	1
PB	1	1	1	1	1	0	1	1	1	0	1	1
PR	1	0	0	0	0	0	0	0	0	0	0	1
HPR	1	0	0	0	0	0	0	0	0	0	0	0
Total	11	2	2	6	2	0	5	2	2	0	5	9
Ranking	1	6	6	3	6	11	4	6	6	11	4	2

Having determined the performance matrices of all the test functions, the best method for each design problem will have a score as 1 whereas the worst has 12 as given in Table 4. After summing up the scores of all the testing functions, the top five EAs and meta-heuristics are: the hybrid PBIL, continuous tabu search, real-code ant colony optimization, simulated annealing, and charged system search. The order of the top five methods is slightly different from that in the first comparison. Among the

PBIL versions, real-code PBIL outperforms binary PBIL, the worst method in this study. The hybrid PBIL is superior to the real-code PBIL, which means the inclusion of an evolutionary direction and an approximate gradient helps enhance PBIL search performance. Charged system search is the best method for larger scale testing functions having one optimum solution while the proposed hybrid approach is the best for multi-modal small scale functions.

Table 4. Comparative results by ranking

Function No.	ACO	CSS	DE	TS	FA	GA	PSO	SA	SS	PB	PR	HPR
1	1	6	6	3	6	11	4	6	6	11	4	2
2	2	9	9	4	6	6	1	5	9	6	9	3
3	3	6	8	2	6	11	3	8	10	11	3	1
4	6	6	6	4	6	4	1	6	6	6	3	1
5	1	8	6	1	4	8	7	8	8	8	1	4
6	2	6	6	3	6	11	3	6	6	12	5	1
7	3	6	8	2	6	9	3	9	9	9	3	1
8	3	3	2	11	5	5	12	1	5	5	5	5
9	4	4	7	2	4	10	7	11	7	11	2	1
10	11	3	9	2	6	9	7	3	8	11	3	1
11	10	5	10	3	4	5	5	2	10	5	5	1
12	2	7	7	5	5	7	7	4	7	7	3	1
13	7	4	12	1	7	7	3	4	7	7	4	1
14	1	12	2	3	3	6	10	10	6	9	8	3
15	1	1	8	5	10	5	4	10	3	12	8	5
16	6	4	9	2	4	11	9	7	7	12	3	1
17	1	12	3	5	4	5	11	1	10	9	5	5
18	11	1	12	7	7	4	7	6	2	3	7	4
19	3	7	1	9	10	4	11	1	6	5	12	7
20	4	1	4	2	4	4	4	12	4	4	4	2
21	3	1	7	5	6	11	9	4	9	12	8	2
22	4	8	1	7	4	6	12	1	11	9	9	1
23	4	1	9	1	5	10	11	3	7	11	8	5
24	3	2	6	11	11	4	8	1	4	7	8	8
25	6	1	9	3	5	10	11	3	6	11	6	2
26	12	5	9	9	6	2	6	1	6	3	11	4
27	4	2	10	1	5	9	10	2	5	10	7	7
28	3	1	8	5	6	9	11	3	9	11	7	2
29	12	3	5	5	5	2	10	1	10	3	9	5
30	2	5	2	7	1	8	12	8	11	8	5	4
31	4	2	8	6	4	8	12	1	10	11	6	2
32	4	1	7	3	7	9	12	2	10	11	6	4
33	1	11	1	1	1	1	1	1	1	1	12	1
34	5	1	8	4	5	9	12	3	10	11	7	2
35	3	1	6	6	4	8	12	2	8	10	11	5
total	152	156	231	150	188	248	268	156	253	292	217	104

5 Conclusions and Discussion

The real-code PBIL is developed to deal with box-constrained optimization. The hybridization of the real-code PBIL with the evolutionary directions and approximate gradient is proposed. The new method is derivative-free and capable of maintaining the outstanding advantages of traditional EAs e.g. global optimization. From the comparative results, it is shown that the proposed hybrid approach is one of the best EAs. In fact, it is the overall best method based on the assessment in this work. The

main real-code PBIL is used for global search while the evolutionary direction and approximate gradient are efficient for local search. Nevertheless, it should be noted that the comparative results rely on specific optimization settings such as crossover and mutation probabilities of GA, and number of fireworks in FA. The future work will be the implementation of an approximate gradient, an evolutionary direction, and some other efficient evolutionary operators for constrained optimization problems, and multiobjective optimization.

Acknowledgments. The author is grateful of the support from the Thailand Research Fund (TRF). Many thanks are also directed to my colleague, Peter Warr, for his careful proofreading.

References

1. Farina, M., Amato, P.: Linked Interpolation-Optimization Strategies for Multicriteria Optimization Problems. *Soft Computing* 9, 54–65 (2005)
2. Srisoporn, S., Bureerat, S.: Geometrical Design of Plate-Fin Heat Sinks Using Hybridization of MOEA and RSM. *IEEE Transactions on Components and Packaging Technologies* 31, 351–360 (2008)
3. Kaveh, A., Talatahari, S.: Particle Swarm Optimizer, Ant Colony Strategy and Harmonic Search Scheme Hybridized for Optimization of Truss Structures. *Computer and Structures* 87, 1245–1287 (2009)
4. Baluja, S.: 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 (1994)
5. Sebag, M., Ducoulombier, A.: Extending Population-Based Incremental Learning to Continuous Search Spaces. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) *PPSN 1998. LNCS*, vol. 1498, pp. 418–427. Springer, Heidelberg (1998)
6. Yuan, B., Gallagher, M.: Playing in Continuous Spaces: Some Analysis and Extension of Population-Based Incremental Learning. In: *CEC 2003*, CA, USA, pp. 443–450 (2003)
7. Bureerat, S., Cooper, J.E.: Evolutionary Optimisation Using Evolutionary Direction and Pseudo-Gradient. In: *1st ASMO UK/ISSMO*, Ilkley, UK, pp. 81–87 (1999)
8. Yamamoto, K., Inoue, O.: New Evolutionary Direction Operator for Genetic Algorithms. *AIAA* 33, 1990–1993 (1995)
9. Lindfield, G., Penny, J.: *Numerical Methods Using MATLAB*. Ellis Horwood, England (1995)
10. Socha, K., Dorigo, M.: Ant Colony Optimization for Continuous Domains. *European Journal of Operational Research* 185, 1155–1173 (2008)
11. Herrera, F., Lozano, M., Molona, D.: Continuous Scatter Search: An Analysis of the Integration of Some Combination Methods and Improvement Strategies. *European Journal of Operational Research* 169, 450–476 (2006)
12. Kaveh, A., Talatahari, S.: A Novel Heuristic Optimization Method: Charged System Search. *Acta Mechanica* 213, 267–289 (2010)
13. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012. International Computer Science Institute, Berkeley, CA (1995)

14. Teh, Y.S., Rangaiah, G.P.: Tabu Search for Global Optimization of Continuous Functions with Application to Phase Equilibrium Calculations. *Computers and Chemical Engineering* 27, 1665–1679 (2003)
15. Tan, Y., Zhu, Y.: Fireworks Algorithm for Optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) *ICSI 2010. LNCS*, vol. 6145, pp. 355–364. Springer, Heidelberg (2010)
16. Reyes-Sierra, M., Coello Coello, C.A.: Multi-objective Particle Swarm Optimizers: a Survey of the State-of-the-Art. *Int. J. of Computational Intelligence Research* 2, 287–308 (2006)
17. Bureerat, S., Limtragool, J.: Structural Topology Optimisation Using Simulated Annealing with Multiresolution Design Variables. *Finite Element in Analysis and Design* 44, 738–747 (2008)

Improved Population-Based Incremental Learning in Continuous Spaces

Sujin Bureerat

Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University,
Thailand, 40002, Email: sujbur@kku.ac.th

Abstract Population-based incremental learning (PBIL) is one of the well-established evolutionary algorithms (EAs). This method, although having outstanding search performance, has been somewhat overlooked compared to other popular EAs. Since the first version of PBIL, which is based on binary search space, several real code versions of PBIL have been introduced; nevertheless, they have been less popular than their binary code counterpart. In this paper, a population-based incremental learning algorithm dealing with real design variables is proposed. The method achieves optimization search with the use of a probability matrix, which is an extension of the probability vector used in binary PBIL. Three variants of the new real code PBIL are proposed while a comparative performance is conducted. The benchmark results show that the present PBIL algorithm outperforms both its binary versions and the previously developed continuous PBIL. The new methods are also compared with well-established and newly developed EAs and it is shown that the proposed real-code PBIL is can rank among the high performance EAs.

Keywords: Population-based incremental learning; evolutionary algorithms; performance comparison; meta-heuristics; continuous domains

1 Introduction

Evolutionary algorithms (EAs) have long been regarded as alternative optimizers, apart from classical mathematical programming. This variety of optimizers is well-known and popular for their robustness, derivative – free searching and capa-

bility in tackling global optimisation. However, these methods have no guarantees of convergence and search consistency. Moreover, they usually require a large number of function evaluations to achieve optimum solutions. With such advantages and disadvantages, there have been numerous evolutionary algorithms and meta-heuristic (MH) searches developed with the aim to enhance search performance whilst maintaining their outstanding abilities e.g. global optimization capability, derivative-free strategy, and robustness.

Methods, such as genetic algorithm (GA), are said to be well-established and have been implemented on a wide range of real world applications. PBIL, on the other hand, has been proposed as an alternative binary code approach to GA. The method has been somewhat overlooked although it has an acceptable search performance when compared to some popular EAs. In this paper, population-based incremental learning dealing with real design variables is proposed. The method achieves optimization search with the use of a probability matrix, which is an extension of a probability vector used in the binary PBIL. Three variants of the new real code PBIL are proposed while a comparative performance is performed. The benchmark results show that the present PBIL algorithm outperforms its binary versions and the previously developed continuous PBIL. The new methods are also compared with well-established EAs, and some other newly developed algorithms. It is shown that the proposed real-code PBIL is considered among the best EAs.

2 PBIL Algorithms

The first version of PBIL is based upon a binary search space [1]. It was proposed as an alternative EA apart from the best known GA. The basic idea of binary PBIL is to use the so-called probability vector to represent a set of design solutions or population rather than using a set of binary design solutions as with GA. During the searching process, the probability vector is improved iteratively until termination conditions are met. This search strategy can be viewed as limiting a binary search space iteratively until reaching the optimum solution. The real-code versions of PBIL have been developed later, e.g. PBIL_I and PBIL_G [2-3]. Moreover, since the introduction of binary PBIL, a number of estimation distribution algorithms have been proposed, such as the bivariate marginal distribution algorithm [4], and multivariate interactions (e.g. Bayesian optimization algorithm [5]). Multiobjective versions of PBIL have also been proposed [6] and implemented on a variety of engineering applications. The most outstanding feature of PBIL, when dealing with a multiobjective problem, is its ability to provide better population diversity. This means that an efficient, approximate, non-dominated set can be expected. The real code PBIL presented in this paper is developed for single objective optimization with box constraints, which can be expressed as:

$$\text{Min } f(\mathbf{x}) \quad (1)$$

$$\begin{aligned} L_i &\leq x_i \leq U_i; i = 1 \dots n \\ \mathbf{x} &\in R^n \end{aligned}$$

where $\mathbf{x}$ is the vector of design variables size $n \times 1$, f is an objective function, L_i are lower bounds of $\mathbf{x}$, and U_i are upper bounds of $\mathbf{x}$.

For binary PBIL, the so-called probability vector, $\mathbf{P}$, is used to create a binary population. Given that $\mathbf{x}$ is represented by a binary string, $\mathbf{b}$, size $N_b \times 1$, the probability vector will have the same size as the binary design solution. The i -th element of the probability vector determines the number of '1' bits on the row of a binary population $\{\mathbf{b}_1 \dots \mathbf{b}_M\}$. If the population size is $N_b \times M$, the i -th row of the population will have approximately $P_i M$ elements of '1' in which their positions are randomly located.

In continuous domains, it is more complicated to represent real number design solutions with a probability vector. From the previous work of continuous PBIL, PBIL_L and PBIL_G use a probability vector, whereas the histogram PBIL_H [3] proposes the use of a probability matrix. In this work, a probability matrix similar to that used in PBIL_H is used; however, the search strategy and probability updating scheme are totally different from PBIL_H. Given that the probability matrix is sized $n \times T$, the range $[L_i, U_i]$ is thus divided in to T equal parts leading to a set of points $\mathbf{a}_i = \{a_0 \dots a_T\}$ where $a_0 = L_i$ and $a_T = U_i$. Each element P_{ij} indicates a number of x_i in a current population, which have values in the j -th interval of $\mathbf{a}_i$. Initially, a matrix $\mathbf{A}$, size $n \times (T+1)$ whose i -th row contains $\mathbf{a}_i$ is created and used together with the probability matrix. The procedure to generate a real code population $\{\mathbf{x}_1 \dots \mathbf{x}_M\}$ is detailed in Algorithm 1 where $\lambda \in [0,1]$ is a uniform random number. The command lines 7-9 use a bias random generation so that y_l has the possibility to be located at the bounds of the sub-interval $[a_{i,j-1}, a_{ij}]$.

The main search procedure of the proposed real code PBIL starts with a probability matrix whose elements are set to be $1/T$. A real number population is then created using Algorithm 1. Having evaluated the objective function values of all the members in the population, the best individual, $\mathbf{x}^{\text{best}}$, is detected and will be used for updating the probability matrix. The probability matrix updating scheme is given in Algorithm 2. Note that mutation in the updating process is used to prevent premature convergence of the real-code PBIL. At the i -th loop, the r -th interval of $\mathbf{a}_i$ which contains x_i^{best} is detected. The concept of updating the i -th row of $\mathbf{P}$ is to increase the value of P_{ir} . The command lines 3-6 are proposed to prevent the premature convergence of the method. The command line 10 is activated so as to fulfill the condition

$$\sum_{j=1}^T P_{ij} = 1. \quad (2)$$

$\text{rand} \in [0,1]$ is a uniform random number generated every time the command line 7 is activated. In this paper, the learning rate (L_{R0}), mutation probability, and mutation shift (M_s) are set to be 0.5, 0.02, and 0.2 respectively.

The probability matrix and the best individual are updated iteratively until the termination condition is met. The procedure of the real-code PBIL is given in Al-

gorithm 3 where N_G is the maximum number of generations assigned for termination condition.

Algorithm 1. Creating real code design solutions from a probability matrix

Input: $P_{ij}, n, T, M, a_{ij} (\mathbf{A})$
Output: $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$
1: For $i = 1$ to n
2: $l = 0$
3: For $j = 1$ to T
4: $N_j = \text{round}(P_{ij} \cdot M)$
5: For $k = 1$ to N_j
6: Set $l = l + 1$
7: Assign $y_l = 0.9a_{i,j-1} + \lambda(1.1a_{ij} - 0.9a_{i,j-1})$
8: If $y_l < a_{i,j-1}$; set $y_l = a_{i,j-1}$
9: If $y_l > a_{ij}$; set $y_l = a_{ij}$
10: If $l = M$, break
11: End
12: End
13: Set the i -th row of $\mathbf{X} = \text{randomly_permute}(y_l), l = 1, \dots, M$
14: End

Algorithm 2. Probability matrix updating scheme

Input: $P_{ij}^{old}, \mathbf{x}^{best}, a_{ij}, L_{R0}$, mutation probability, M_s
Output: P_{ij}^{new}
1: For $i = 1$ to n
2: Search for the r -th interval of $\mathbf{a}_i$, which contains x_i^{best}
3: For $j = 1$ to T
4: $L_R = L_{R0} \exp(-(j - r)^2)$
5: $P'_{ij} = (1 - L_R)P_{ij}^{old} + L_R$
6: End
7: If $rand < \text{mutation probability}$
8: Select $t \in \{1, \dots, T\}$ randomly
9: $P'_{it} = (1 - M_s)P'_{it} + \text{round}(rand)M_s$
10: Normalize $\mathbf{P}$:

$$P''_{ij} = \frac{1}{\sum_{j=1}^T P'_{ij}} P'_{ij}$$

11: $P_{ij}^{new} = P''_{ij}$
12: End

Algorithm 3. Real-Code Population-Based Incremental Learning

Input: N_G, M, n, T, L_{R0} , mutation probability, M_s , function name (fun)
Output: $\mathbf{x}^{best}, f^{best}$

Initialization: $P_{ij} = 1/T$, a_{ij} , $\mathbf{x}^{\text{best}} = \{\}$
 1: For $i = 1$ to N_G
 2: Generate a real code population $\mathbf{X}$ from P_{ij} , a_{ij} (**Algorithm 1**) and find $\mathbf{f} = \text{fun}(\mathbf{X})$
 3: Find new $\mathbf{x}^{\text{best}}$ from $\mathbf{X}$ and $\mathbf{x}^{\text{best}}$ from the previous generation
 4: Update P_{ij} based on the current $\mathbf{x}^{\text{best}}$ (**Algorithm 2**)
 5: End

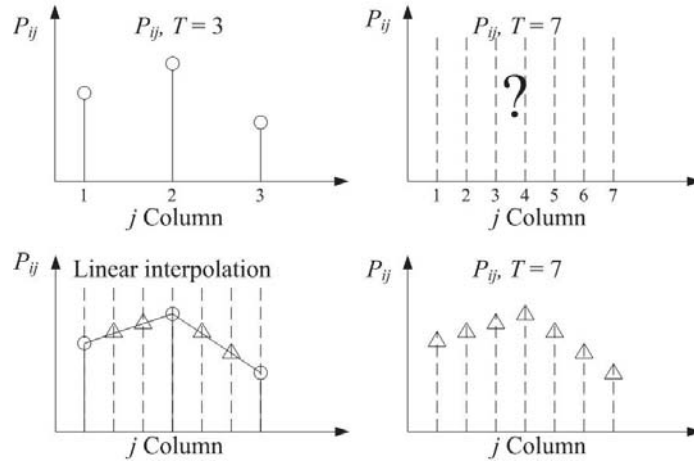


Fig. 1. Transformation of probability matrices: Δ larger T (fine), $\circ$ smaller T (coarse)

One of the parameters that influence the search performance of the proposed continuous PBIL is the number of columns of the probability matrix denoted by T . The coarser probability matrix (with lower value of T) is suitable for global search whereas the finer probability matrix is suitable for local search. In this paper, three variants of real-code PBIL are proposed as follows:

RPBIL1 is a real-code PBIL using a constant size probability vector (default $T = 30$).

RPBIL2 is a real-code PBIL using two probability matrices with different sizes. The coarse probability matrix is used for global search while, at the same time, the finer probability matrix is used for local search (default $T_{\text{coarse}} = 10$, $T_{\text{fine}} = 30$). Each probability matrix is used to generate half of the solutions in a population.

RPBIL3 is a real-code PBIL using an adaptive probability vector. The strategy is to start with a coarser probability matrix at the early iterations. As the algorithm proceeds, the value of T becomes larger. The transformation from coarser to finer probability matrices can be carried out by using linear interpolation. Fig. 1 displays the transformation from a particular row of a coarser matrix ($T = 3$) to a row of a finer matrix ($T = 7$). The marker “ $\circ$ ” represents the values of the coarser matrix elements whereas the marker “ Δ ” represent the values of the finer matrix elements. In this work, we use $T_{\text{start}} = 5$, and $T_{\text{end}} = 40$ where T reaches T_{end} at the half way point of an optimization run and afterwards remains constant until the search is terminated.

3 Performance Testing

Measurement of the search performance of the proposed PBILs is carried out by using a number of testing objective functions. The computation results obtained from using PBILs will be compared with those obtained from using some well-established and newly developed EAs. The implemented EAs are: binary-code genetic algorithm (GA) [7], stud-genetic algorithm (SGA) [8], real-code ant colony optimization (ACO) [9], continuous scatter search (SS) [10], particle swarm optimization (PSO) [11], charged system search (CSS) [12], fireworks algorithm (FA) [13], differential evolution (DE) [14], covariance matrix adaptation evolution strategy (CMA-ES) [15], binary PBIL [1], PBIL_G [3], RPBIL1, RPBIL2, and RPBIL3. The PBIL_H, although being a predecessor closest to the proposed PBIL, will not be implemented because it uses all of the solutions in a population to update the probability matrix, which means it is time-consuming. Moreover, all the objective function values need to be positive or converted to be positive before entering an optimization process. This makes the algorithm difficult to use. The testing functions are detailed in Table 1. Note that function expressions may not be shown herein as we have limited paper length. The functions are taken from [9] and [10] where they are traceable from their names. For each test function, all evolutionary algorithms start with the same initial population and search for the optimum 30 simulation runs. The number of iterations is set to be $10n$ while the population size is set to be $7n$, which means the optimization procedures are terminated after $10n \times 7n$ function evaluations. It should be noted that, from observation, most of the algorithms can usually locate near optimum areas in a few early iterations. Later on, they struggle to hit the real optima. Therefore, the assigned generation number and population size used in this paper are adequate for EAs to locate near optimum regions of unconstrained optimization problems rather than to drive EAs to strict optimum points. The best results the optimizers can explore in the last generation are taken as optimum results. Several methods, such as SS and FA, due to their search mechanisms, may not use the same number of generations and population size; however, they are assigned to exploit the same number of function evaluations as the others.

Table 1. Testing functions

Function no., details	$[L, U]^n$	Function no., details	$[L, U]^n$
1, B_2 [9]	$[-50, 100]^2$	18, Griewangk [9]	$[-5.12, 5.12]^{10}$
2, Beale [10]	$[-4.5, 4.5]^2$	19, $Perm$ [10]	$[-15, 15]^{15}$
3, Booth [10]	$[-10, 10]^2$	20, $Perm0$ [10]	$[-15, 15]^{15}$
4, Easom [9]	$[-100, 100]^2$	21, Cigar [9]	$[-3, 7]^{20}$
5, Goldstein & Price [9]	$[-2, 2]^2$	22, Diagonal plane [9]	$[0.5, 1.5]^{20}$
6, Martin & Gaddy [9]	$[-20, 20]^2$	23, Dixon & Price [10]	$[-10, 10]^{20}$
7, Matyas [10]	$[-5, 10]^2$	24, Levy(n) [10]	$[-10, 10]^{20}$
8, Penny & Linfield* [7]	$[-5, 5]^2$	25, Powell(n) [10]	$[-4, 5]^{20}$
9, Powersum [10]	$[0, 2]^2$	26, Rastrigin(n) [9]	$[-2.56, 5.12]^{20}$
10, Branin [9]	$[-5, 15]^2$	27, Rosenbrock [9]	$[-5, 5]^{20}$
11, Shubert [10]	$[-10, 10]^2$	28, Sum Squares [10]	$[-5, 10]^{20}$
12, Six Hump		29, Schwefel [10]	$[-500, 500]^{20}$

Camel Back [10]	$[-5,10]^2$	30, Trid(n) [10]	$[-400,400]^{20}$
13, Colville [10]	$[-10,10]^4$	31, Ackley(n) [10]	$[-15,30]^{30}$
14, Hartmann 3,4 [9]	$[0,4]^4$	32, Ellipsoid [9]	$[-3,7]^{30}$
15, Shekel [9]	$[0,10]^4$	33, Plane [9]	$[-0.5,1.5]^{30}$
16, Zakharov [9]	$[-5,10]^5$	34, Sphere [9]	$[-3,7]^{30}$
17, Hartmann 6,4 [9]	$[0,6]^6$	35, Tablet [9]	$[-3,7]^{30}$

$$* f_8(\mathbf{x}) = 0.5(x_1^4 - 16x_1^2 + 5x_1) + 0.5(x_2^4 - 16x_2^2 + 5x_2)$$

4 Comparative Results

Having performed 14 optimizers to solve 35 unconstrained optimization problems, the comparative results are given in Table 2. Each value in the table is the average of 30 near optimum values each method can search for. The average values are normalized with respect to the best and the worst methods, which can be expressed as:

$$\bar{f}_i = \left| \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right| \quad (3)$$

where $f_{\min}$ is the average near optimum value of the best method, and $f_{\max}$ is the average near optimum value of the worst method. By using Eq. 3, the best method will have $\bar{f} = 0$ whereas the worst method will have $\bar{f} = 1$.

From the overall comparison results, the five best methods are RPBIL3, ACO, FA, GA, and DE. For the functions F_{20} and F_{33} , most of the optimizers can reach the real optima. Between the top two optimizers, RPBIL3 is better when the design variables vectors have smaller size while ACO is slightly superior when dealing with design variables with larger scale. However, the best method for large-scale problems in this performance test is CSS. Among the PBIL variants, RPBIL3 is the best while the second best is RPBIL1. The third best is the original binary PBIL whereas the worst is RPBIL2. Based on the standard deviations of the near optimum values of the various optimizers which are not shown in the paper, all of the algorithms do not have an acceptable search consistency. This could be due to the low number of generations and small population size being used.

Table 2. Comparative results by normalized function values

Fn No.	ACO	CSS	FA	GA	PSO	SGA	SS	DE	ES	PB	PG	PR1	PR2	PR3
1	0	0.007	0.016	0.507	0	1	0.018	0.01	0	0.429	0.032	0	0.043	0
2	0.027	1	0.395	0.189	0	0.425	0.175	0.467	0.17	0.298	0.52	0.1	0.121	0.085
3	0.004	0.048	0.021	0.367	0	1	0.135	0.159	0	0.214	0.066	0.023	0.575	0
4	0.977	1	1	0.622	0.522	0.999	1	0.993	1	0.886	0.559	1	1	0
5	0	0.712	0.036	0.358	0.181	0.92	0.527	0.242	0.333	0.295	0.084	0.11	1	0.108
6	0.001	0.023	0.02	0.174	0.002	1	0.073	0.066	0	0.589	0.018	0.004	0.315	0
7	0.034	0.067	0.085	0.375	0.02	0.585	0.515	0.228	0	1	0.926	0.073	0.146	0.001
8	0.038	0.16	0.336	0.091	1	0.193	0.092	0.031	0.545	0.128	0.032	0.71	0.007	0

9	0.009	0.009	0.063	0.191	0.086	1	0.082	0.096	0	0.571	0.073	0.009	0.456	0.009
10	0.382	0.018	0.042	0.135	0.285	1	0.15	0.368	0.603	0.319	0.288	0.019	0.366	0
11	0.459	0.295	0.103	0.194	0.265	0.199	0.506	0.416	1	0.126	0.244	0.123	0.07	0
12	0.025	0.151	0.083	0.156	0.173	0.391	0.255	0.09	0	0.196	0.094	1	0.144	0.027
13	0.077	0.062	0.118	0.083	0.052	0.075	0.078	0.211	0	0.082	1	0.045	0.13	0.056
14	0	1	0.004	0.036	0.336	0.37	0.051	0.001	0.129	0.091	0.168	0.144	0.221	0.011
15	0	0.098	0.867	0.399	0.542	1	0.186	0.687	0.439	0.962	0.822	0.661	0.342	0.027
16	0.026	0.007	0.02	0.473	0.262	1	0.188	0.232	0	0.742	0.262	0	0.635	0.005
17	0	1	0.102	0.194	0.904	0.312	0.524	0.047	0.911	0.328	0.192	0.263	0.102	0.26
18	0.502	0	0.099	0.062	0.083	0.103	0.01	0.589	1	0.044	0.317	0.058	0.715	0.028
19	0	0.447	0.585	0.032	0.8	0	0.24	0	0.201	0.059	0.341	1	0.3	0.956
20	0	0	0	0	0	0.172	0	0.001	0	0	0	0	1	0
21	0	0	0.005	0.054	0.04	0.1	0.032	0.006	0	0.186	0.737	0.01	1	0.007
22	0	0.015	0	0.001	1	0	0.057	0	0	0.022	0	0.057	0.035	0.053
23	0	0	0	0.001	0.003	0.001	0	0.001	0	0.003	0.041	0	1	0
24	0.007	0	0.31	0.013	0.16	0.112	0.012	0.042	0.005	0.119	0.209	0.085	1	0.091
25	0.003	0	0.001	0.016	0.021	0.047	0.002	0.007	0	0.057	0.152	0.006	1	0.003
26	0.886	0.259	0.371	0	0.319	0.26	0.352	0.462	0.087	0.114	1	0.615	0.678	0.424
27	0.003	0.002	0.002	0.005	0.021	0.008	0.003	0.017	0	0.017	0.088	0.014	1	0.005
28	0	0	0.006	0.052	0.218	0.308	0.04	0.012	0	0.245	1	0.012	0.783	0.01
29	0.955	0.109	0.394	0	0.716	0.031	0.789	0.402	0.63	0.107	0.832	0.603	1	0.435
30	0.124	0.289	0	0.384	1	0.512	0.619	0.071	0.153	0.455	0.897	0.442	0.644	0.329
31	0.202	0.109	0.187	0.363	0.663	0.646	0.447	0.359	0	0.543	1	0.287	0.941	0.271
32	0.002	0	0.016	0.047	0.354	0.118	0.056	0.014	0	0.172	1	0.046	0.552	0.018
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0.003	0	0.004	0.045	0.405	0.457	0.079	0.026	0	0.136	1	0.019	0.702	0.011
35	0	0	0.001	0.018	0.932	0.125	0.024	0.004	0	0.053	0.448	1	0.423	0.722
total	4.746	6.887	5.292	5.637	11.36	14.46	7.317	6.357	7.206	9.588	14.44	8.538	18.44	3.952

* PB = binary PBIL, PG = PBILG, PR1 = RPBIL1, PR2 = RPBIL2, PR3 = RPBIL3, ES = CMAES

Table 3 provides an alternative comparison which is based on a ranking approach. For each testing function, the 12 optimizers are ranked based on their average values in Table 2, where the better methods have smaller scores. Based on the total scores, the top five optimizers are still the same as those ranked by objective function values with a slightly different order as ACO, CMAES, RPBIL3, CSS, and FA. Other conclusions are similar to the comparison in Table 2.

Table 3. Comparative results by ranking

Fn No.	ACO	CSS	FA	GA	PSO	SGA	SS	DE	ES	PB	PG	PR1	PR2	PR3
1	1	6	8	13	1	14	9	7	1	12	10	1	11	1
2	2	14	10	8	1	11	7	12	6	9	13	4	5	3
3	4	7	5	12	1	14	9	10	1	11	8	6	13	1
4	6	9	9	4	2	8	9	7	9	5	3	9	9	1
5	1	12	2	10	6	13	11	7	9	8	3	5	14	4
6	3	8	7	11	4	14	10	9	1	13	6	5	12	1
7	4	5	7	10	3	12	11	9	1	14	13	6	8	2
8	5	9	11	6	14	10	7	3	12	8	4	13	2	1
9	2	2	6	11	9	14	8	10	1	13	7	2	12	2
10	12	2	4	5	7	14	6	11	13	9	8	3	10	1
11	12	10	3	6	9	7	13	11	14	5	8	4	2	1
12	2	8	4	9	10	13	12	5	1	11	6	14	7	3
13	7	5	11	10	3	6	8	13	1	9	14	2	12	4

14	1	14	3	5	12	13	6	2	8	7	10	9	11	4
15	1	3	12	6	8	14	4	10	7	13	11	9	5	2
16	6	4	5	11	9	14	7	8	1	13	9	1	12	3
17	1	14	3	6	12	9	11	2	13	10	5	8	3	7
18	11	1	8	6	7	9	2	12	14	4	10	5	13	3
19	1	10	11	4	12	1	7	1	6	5	9	14	8	13
20	1	1	1	1	1	13	1	12	1	1	1	1	14	1
21	1	1	4	10	9	11	8	5	1	12	13	7	14	6
22	1	8	1	7	14	1	12	1	1	9	1	12	10	11
23	1	1	1	8	11	8	1	8	1	11	13	1	14	1
24	3	1	13	5	11	9	4	6	2	10	12	7	14	8
25	5	1	3	9	10	11	4	8	1	12	13	7	14	5
26	13	4	8	1	6	5	7	10	2	3	14	11	12	9
27	4	2	2	6	12	8	4	10	1	10	13	9	14	6
28	1	1	4	9	10	12	8	6	1	11	14	6	13	5
29	13	4	5	1	10	2	11	6	9	3	12	8	14	7
30	3	5	1	7	14	10	11	2	4	9	13	8	12	6
31	4	2	3	8	12	11	9	7	1	10	14	6	13	5
32	3	1	5	8	12	10	9	4	1	11	14	7	13	6
33	1	2	3	4	5	6	7	8	9	10	11	12	13	14
34	3	1	4	8	11	12	9	7	1	10	14	6	13	5
35	1	1	4	6	13	9	7	5	1	8	11	14	10	12
total	140	179	191	251	291	348	269	254	156	319	340	242	376	164

5 Conclusions and Discussion

Three versions of new real-code population-based incremental learning are proposed. The methods use a probability matrix to represent a set of real design variables during the search procedure. The comparative results show that the proposed real-code PBIL is comparable to high performance evolutionary algorithms. It should be noted that a more reliable comparison should be made based on a statistical approach rather than by comparing the average objective values directly. Since RPBIL3 outperforms the other PBIL versions, it can be concluded that the probability matrix should start with a low resolution at early iterations. This will help the PBIL to perform a global search. The probability matrix resolution becomes higher in the later iterations so as to make a fine search for the real optimum. For the other versions, RPBIL1, which uses one fine probability matrix, is superior to RPBIL2, which uses both coarse and fine probability matrices at the same time. This result is due to use of the finer probability matrix tends to produce a better search performance than the use of the coarser one. Since RPBIL2 uses the coarser probability matrix to produce half of the solution members throughout the simulation run, it therefore has an inferior performance when compared to the other two RPBIL variants. The proposed real-code PBIL, with adaptive probability matrix, is an algorithm that performs among the best EAs. It is possible to enhance its search performance by hybridizing it with some other efficient evolutionary operators.

Acknowledgments The corresponding author is grateful of the support from the Thailand Research Fund (TRF). Many thanks are also directed to my colleague, Peter Warr, for making this paper more readable.

References

1. Baluja, S.: 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 (1994)
2. Sebag, M., Ducoulombier, A.: Extending Population-Based Incremental Learning to Continuous Search Spaces. LNCS, 1498 (1998) 418-427
3. Yuan, B., Gallagher, M.: Playing in Continuous Spaces: Some Analysis and Extension of Population-Based Incremental Learning. In: CEC2003, CA, USA, (2003) 443-450
4. Pelikan, M., Muhlenbein, H.: The Bivariate Marginal Distribution Algorithm. *Advances in Soft Computing - Engineering Design and Manufacturing*, (1999) 521-535
5. Pelikan, M., Goldberg, D.E., Cantú-Paz, E.: BOA: The Bayesian Optimization Algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, I (1999) 525-532
6. Bureerat, S., Sriwornamas, K.: Population-Based Incremental Learning for Multiobjective Optimisation. *Advances in Soft Computing*, 39 (2007) 223-232
7. Lindfield, G., Penny, J.: *Numerical Methods Using MATLAB*, Ellis Horwood, (1995)
8. Khatib, W., Fleming, P.: The Stud GA: A Mini-Revolution. In: *5th Int. Conf. on Parallel Problem Solving From Nature*, (1998)
9. Socha, K., Dorigo, M.: Ant Colony Optimization for Continuous Domains. *European Journal of Operational Research*, 185 (2008) 1155-1173
10. Herrera, F., Lozano, M., Molina, D.: Continuous Scatter Search: An Analysis of the Integration of Some Combination Methods and Improvement Strategies. *European Journal of Operational Research*, 169 (2006) 450-476
11. Reyes-Sierra, M., Coello Coello, C.A.: Multi-objective Particle Swarm Optimizers: a Survey of the State-of-the-Art. *Int. J. of Computational Intelligence Research*, 2(3) (2006) 287-308
12. Kaveh, A., Talatahari, S.: A Novel Heuristic Optimization Method: Charged System Search. *Acta Mechanica*, 213(3-4) (2010) 267-289
13. Tan, Y., Zhu, Y.: Fireworks Algorithm for Optimization. *Advances in Swarm Intelligence*, 6145/2010 (2010) 355-364
14. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11 (1997) 341-359
15. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1) (2003) 1-18

Passive vibration control of an automotive component using evolutionary optimisation

Nantiwat Pholdee¹ and Sujin Bureerat²

^{1,2}Department of Mechanical Engineering, Faculty of Engineering, Khon Kaen University,
Khon Kaen, Thailand 40002

Abstract

In this paper, the use of multiobjective evolutionary optimisers for passive vibration suppression of an automotive component is demonstrated. The component is used to connect a car engine to some point of a car body between the front seats. Under such a circumstance, the structure is subject to several mechanical phenomena e.g. stress failure, fatigue, vibration resonance, and vibration transmissibility. The optimisation problem is posed to find structural shape and size such that maximising structural natural frequency and simultaneously minimising structural mass while constraints include stress failure and displacement. The multiobjective optimiser employed is the multiobjective version of Population-Based Incremental Learning (PBIL) with and without using a surrogate model. The optimum results obtained are illustrated and discussed. It is found that the proposed design scheme is effective and efficient for an automotive component design.

Keywords: multiobjective evolutionary algorithm; shape optimisation; Pareto optimal front; automotive component; Vibration suppression

1. Introduction

Due to highly increasing competitiveness in automotive industry, many car manufacturers require to develop new products to offer to customers. Therefore, automotive components are always improved by means of design optimisation [1-2].

Practical engineering design problems are usually assigned to find the best solutions of design variables that lead to optimised design objectives whilst fulfilling all the predefined constraints. Often, the design problem has more than one objective which is called multiobjective optimisation. The most popular method used for the multiobjective optimisation is Evolutionary Algorithms (EAs) [3-6]. The method can explore a Pareto optimum front within a single run and without requiring function derivatives. However, a lack of search consistency and low convergence rate are the inevitable drawbacks of the multiobjective evolutionary algorithms (MOEAs) [5]. For this reason, the hybridisation of a surrogate model method and multiobjective optimisers has been invented and this approach is found to be very powerful and effective [6].

This paper presents the multiobjective evolutionary optimisation of an automotive component. The component is used to connect a car engine to some point of the body between the front seats. The structure is subject to several mechanical phenomena such as stress failure, fatigue, vibration resonance, and vibration transmissibility. The design problem is posed to find structural shape and size such that maximising structural dynamic stiffness while, at the same run, minimising structural mass. Design constraints include stress and displacement. Three dimensional finite element analysis (FEA) is employed to evaluate the objective and constrain function values. The optimum solutions called Pareto solutions are explored by using PBIL incorporating with a Gaussian process surrogate model and a Latin Hypercube Sampling technique. The proposed design approach is found to be numerically powerful and effective.

2. Surrogate model method

The term 'surrogate model' used in an optimisation process is an approximate model which is used to approximate the objective and constrain functions in optimisation problems [7]. Such a design strategy is useful when dealing with optimisation problems with expensive function evaluation, limited function values available, and problems that need to perform an experiment to evaluate their function values. The hybrid of the surrogate model with an optimiser can be achieved in several ways. One of the commonly used strategies is that, during the main optimisation process, some design solutions have been evaluated. Those solutions and their corresponding objective and constraint values are used to build a surrogate model. This model is then used as an approximate function evaluation. The optimisation with the surrogate model is performed with significantly less running time when compared with using the actual function evaluation. The obtained optimum solution of this design phase is brought to the main optimisation process where its actual function value is determined. With a highly accurate surrogate model, this design strategy is far superior to purely using an evolutionary algorithm. The computational steps are repeated until the termination conditions are fulfilled. The commonly used surrogate models for optimisation are Kriging model [8], radial basis interpolation [6], polynomial interpolation [9] and neural network [10]. In this paper, only the Kriging model is employed.

2.1. Kriging Model

A Kriging model (also known as a Gaussian process model) used herein is the famous MATLAB toolbox named Design and Analysis of Computer Experiments (DACE) [8]. The estimation of function can be thought of as the combination of global and local approximation models i.e.

*Corresponding Author: E-mail: sujbur@kku.ac.th

$$y(\mathbf{x}) = \tilde{f}(\mathbf{x}) + Z(\mathbf{x}) \quad (1)$$

where $\tilde{f}(\mathbf{x})$ is a global regression model, $Z(\mathbf{x})$ is a stochastic Gaussian process with zero mean and non-zero covariance representing a localised deviation, and $\mathbf{x}$ is a design variables vector. In this work, a linear function is used for a global model, which can be expressed as:

$$\tilde{f} = \beta_0 + \sum_{i=1}^n \beta_i x_i = \boldsymbol{\beta}^T \mathbf{f} \quad (2)$$

where $\boldsymbol{\beta} = [\beta_0, \dots, \beta_n]^T$, $\mathbf{f} = \mathbf{f}(\mathbf{x}) = [1, x_1, x_2, \dots, x_n]^T$. The covariance of $Z(\mathbf{x})$ is expressed as:

$$\text{Cov}(Z(\mathbf{x}^p), Z(\mathbf{x}^q)) = \sigma^2 \mathbf{R}[\mathbf{x}^p, \mathbf{x}^q] \quad (3)$$

for $p, q = 1, \dots, N$ where $\mathbf{R}$ is the correlation function between any two of the N design points, and $\mathbf{R}$ is the symmetric correlation matrix size $N \times N$ with the unity diagonal [8]. The correlation function used in this paper is

$$\mathbf{R}(\mathbf{x}^p, \mathbf{x}^q) = \exp(-(\mathbf{x}^p - \mathbf{x}^q)^T \boldsymbol{\theta} (\mathbf{x}^p - \mathbf{x}^q)) \quad (4)$$

where $\boldsymbol{\theta}_i$ are the unknown correlation parameters to be determined by means of the maximum likelihood method. Having found $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, the Kriging predictor can be achieved as

$$\bar{y} = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \mathbf{r}^T(\mathbf{x}) \mathbf{R}^{-1}(\mathbf{y} - \mathbf{F} \boldsymbol{\beta}) \quad (5)$$

Where $\mathbf{F} = [\mathbf{f}(\mathbf{x}^1), \mathbf{f}(\mathbf{x}^2), \dots, \mathbf{f}(\mathbf{x}^N)]^T$ and $\mathbf{r}^T(\mathbf{x}) = [\mathbf{R}(\mathbf{x}, \mathbf{x}^1), \mathbf{R}(\mathbf{x}, \mathbf{x}^2), \dots, \mathbf{R}(\mathbf{x}, \mathbf{x}^N)]$. For more details, see [8].

3. Multiobjective Population-Based Incremental Learning (MOPBIL)

PBIL algorithm is an evolutionary optimiser based upon binary searching space. The PBIL approach evolves its population based upon the so-called probability vector, the probability of having '1' elements on each column of a binary population. The example of how the probability vector works is shown in Fig.1 which implies that one probability vector can produce a variety of binary populations.

In the multiobjective optimisation, more probability vectors should be used in order to obtain a more diverse population; therefore, it is called a probability matrix. Starting with an initial probability matrix that have all elements as "0.5", and an initial Pareto archive, the binary population according to the initial probability matrix is then created. The binary population is decoded and objective values are evaluated. The best binary solutions, whether it is based on minimisation or maximisation, is chosen to update the probability vector $P_{i,j}^{new}$ for the next iteration using the relation

$$P_{i,j}^{new} = P_{i,j}^{old} (1 - L_R) + b_j L_R \quad (6)$$

where L_R is called the learning rate, a value between 0 and 1, to be defined and b_j is the mean value of the j^{th} column of the randomly selected non-dominated binary solutions. For this study, L_R is set as:

$$L_R = 0.5 + \text{rand} \times (0.1 \text{ or } -0.1) \quad (7)$$

where $\text{rand} \in [0, 1]$ is a uniform random number. Mutation on the i^{th} row of the probability matrix is allowed to take place by a predefined probability and it can be expressed as:

$$P_{i,j}^{new} = P_{i,j}^{old} (1 - m_s) + \text{rand} (0 \text{ or } 1) m_s \quad (8)$$

where m_s is the amount of shift used in the mutation.

population 1	population 2	population 3
0 0 1 1	0 1 1 0	0 1 0 1
1 1 0 0	1 1 0 0	1 0 0 1
0 0 1 1	1 0 1 0	0 0 0 1
1 1 0 0	0 0 0 1	0 1 0 0
Probability Vectors		
[0.5, 0.5, 0.5, 0.5]	[0.5, 0.5, 0.5]	[0.25, 0.5, 0.0, 0.75]

Fig.1 Probability vector and their corresponding populations

The updating process is completed when all rows of the probability matrix are changed. The probability matrix is updated and the external Pareto archive is improved iteratively until convergence is achieved.

In cases where the total number of the non-dominated solutions is greater than the archive size, the archiving operator called the normal line method [4] is activated to remove some solutions from the archive. The archiving technique is used to prevent excessive use of computer memory during an optimisation process. The basic idea of the normal line technique is used to remove some non-dominated design solutions while maintaining population diversity in the archive. For more details of multiobjective PBIL, see [11].

4. Design Problem

This paper presents a multiobjective optimisation design problem for an automotive part as shown in Fig. 2. The component is used to connect the car engine with the car body. Under the working conditions, this structure is subject to several mechanical phenomena e.g. stress, fatigue, vibration resonance, and dynamic force transmissibility. Also, the structural displacement due to a number of loading conditions should not exceed the predefined limit.

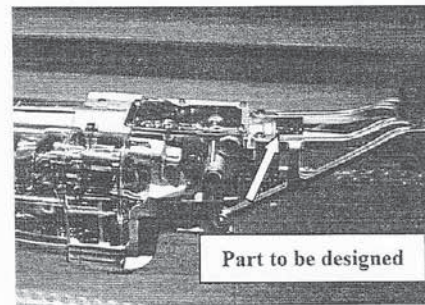


Fig. 2 Automotive part

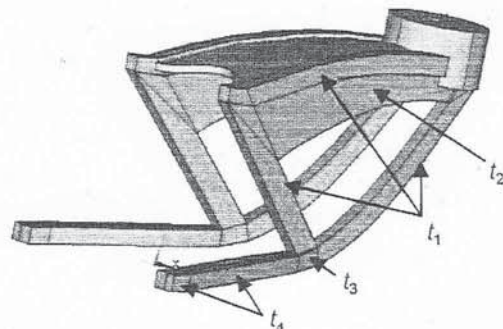


Fig. 3 a. Sizing variables

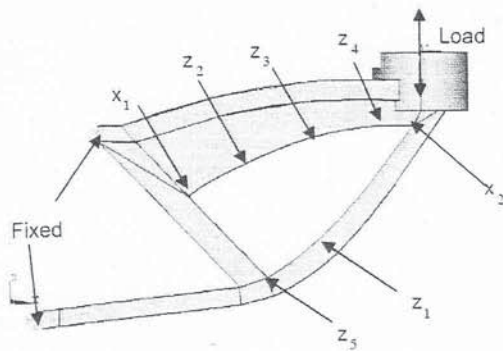


Fig. 3 b. Shape variables

The multiobjective optimisation problem is posed to find structural shape and size such that maximising structural natural frequencies and minimising mass whereas constraint include stress failure and displacement, which can be expressed as

$$\text{Min: } \mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x})] \quad (9)$$

Subject to

$$\sigma_{\max} \leq \sigma_{\text{allowable}}$$

$$u_{\max} \leq 0.005$$

$$0.0015 \leq t_1 \leq 0.0115$$

$$0.0015 \leq t_2 \leq 0.0115$$

$$0.0015 \leq t_3 \leq 0.01$$

$$0.0025 \leq t_4 \leq 0.015$$

$$-0.003 \leq z_1 \leq 0.01$$

$$-0.0025 \leq z_2 \leq 0.0028$$

$$-0.01 \leq z_3 \leq 0.005$$

$$-0.01 \leq z_4 \leq 0.005$$

$$0 \leq z_5 \leq 0.03$$

$$-0.01 \leq x_1 \leq 0.01$$

$$-0.005 \leq x_2 \leq 0.01$$

where $\mathbf{x}$ is a design variable vector (all variables are displayed in Fig. 3). f_1 is a function of mass. f_2 is a function of dynamic stiffness (or natural frequencies). $\mathbf{x}$, f_1 and f_2 can be express as :

$$\mathbf{x} = \{t_1, t_2, t_3, t_4, z_1, z_2, z_3, z_4, z_5, x_1, x_2\}^T$$

and

$$f_1 = \text{mass} \quad (10)$$

and

$$f_2 = \frac{1}{\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5} \quad (11)$$

The other parameters are defined as follows:

$\sigma_{\max}$ = Maximum von Misses stress

$\sigma_{\text{allowable}}$ = Allowable stress

t_i = Shape thickness

z_i = Position of the key points in z -axis direction

x_i = Position of key point in x -axis direction

ω_i = mode i^{th} natural frequency of a structure

Figs. 3a. & 3b. display all of the sizing and shape design variables. The thicknesses (t_i in Fig. 3 a.) are the thickness of the sub-regions of the automotive component as shown. The z_i parameters determine the key points in vertical direction as located in Fig. 3 b. These key points are used to generated a spline curve so as to define the

shape of the part. The x_i parameters define the horizontal position of the key points on the component.

The structure is acted upon by three load cases (bending, twisting and swaying loads) at the right-hand cylinder part. The objective and constraint function values are evaluated by using FEA. The evaluation process is carried out in such a way that, with the given input design variables as defined, the shape and dimensions of the structure are created. The finite element analysis is then performed. Finally the computational results can be obtained. Function evaluation is somewhat time-consuming, which means it is difficult to apply a common evolutionary algorithm to solve the optimisation problem (9). As a result, the surrogate-assisted evolutionary algorithm is developed to deal with such a difficulty.

To tackle multiobjective optimisation as defined in (9), the MOPBIL algorithm and the surrogate-assist MOPBIL (MOPBIL-SM) are used to find Pareto optimal solutions. MOPBIL-SM is a design strategy that exploits the surrogate model to create an initial Pareto archive rather than starting with a randomly generated population as with the traditional multiobjective PBIL.

The computational steps for generating an initial Pareto archive by using the surrogate model are as follows:

- I. Sample a set of design variable vectors from design experiment by using the LHS technique.
- II. Evaluate design functions by FEA.
- III. Constructing a surrogate model by using the Kriging technique.
- IV. Use MOPBIL find Pareto optimal set based on the surrogate model.
- V. Find the real function values of the Pareto optimal front obtained from optimising the approximate Kriging model (step IV).
- VI. Use a non-dominated sorting technique to find the initial Pareto archive

The LHS is used to sample 100 design solutions for constructing a surrogate optimisation model. Subsequently, with this initial Pareto archive, the common MOPBIL is operated where the population size is 30, the number of iterations is 10, and archive size is set as 30.

5. Results and Discussion

The progress of Pareto optimal solutions of the optimisation design problem by using the hybridisation of a surrogate model method and the MOPBIL is displayed in Fig. 4. It can be seen that the Pareto front from iteration 1 to iteration 10 has slight improvement. This means that the initial front generated by means of a surrogate-assisted approach is very powerful.

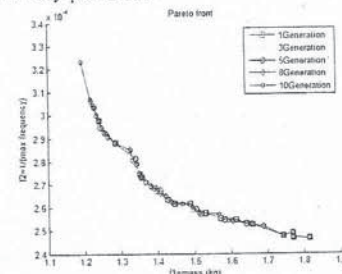


Fig.4 Pareto front of the MOPBIL-SM

In order to verify the effectiveness of the hybrid approach, the original MOPBIL without the use of a surrogate approach is performed with the same population and archive sizes while the total generation number is set to

be 30. This implies that the original MOPBIL uses 30×30 actual function evaluations which is approximately twice the number of evaluation used by MOPBIL-SM ($100 + 10 \times 30$ evaluations). The results from the former are termed as MOPBIL whereas the results obtained from the later are named MOPBIL-SM. Figs.5-7 compare the Pareto fronts obtained from using MOPBIL-SM at the generations of 1, 3 and 5, and using MOPBIL at the generations of 10, 20 and 30 respectively. It can be found that the results from using MOPBIL-SM are better than those obtained from using the original MOPBIL even with a far smaller number of finite element analyses. That means the hybrid approach is far superior to the original optimiser.

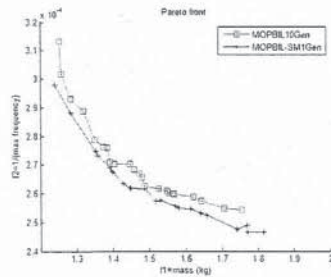


Fig.5 Comparative Pareto fronts: MOPBIL 10 Generations versus MOPBIL-SM 1 Generation

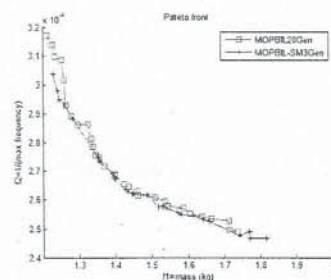


Fig.6 Comparative Pareto fronts: MOPBIL 20 Generations versus MOPBIL-SM 3 Generations

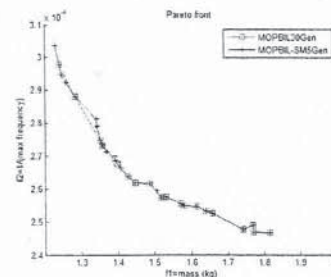


Fig.7 Comparative Pareto fronts: MOPBIL 30 Generations versus MOPBIL-SM 5 Generations

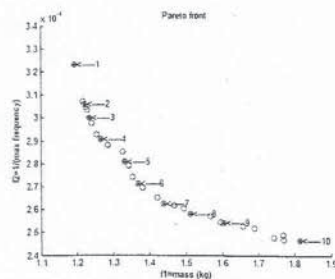


Fig.8 Pareto front from MOPBIL-SM

The Pareto optimal solutions of the MOPBIL-SM shown in Fig. 8 have the corresponding design solutions as shown in Fig. 9. The optimum components have an obvious variation for the design variables t_2 , z_3 and z_5 , while the other variables have a slight variation. It can be seen that, with one optimisation run, we can have a number of optimum components for decision making.

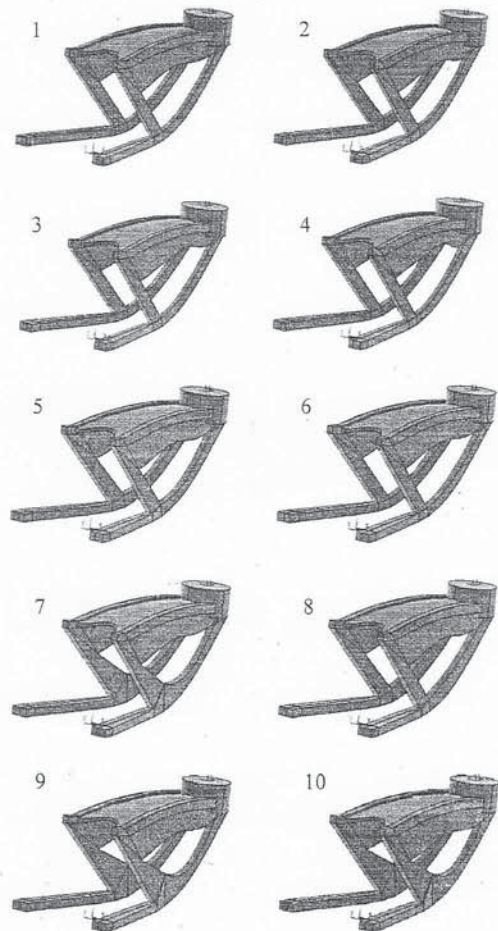


Fig.9 3D automotive parts corresponding to selected solutions in Fig. 8

6. Conclusions

The multiobjective 3D shape and sizing optimisation problem of an automotive component using the hybridisation of a surrogate Kriging model and MOPBIL is demonstrated. The results show that the proposed approach is efficient and effective for solving the design problem. The new design strategy outperforms the original PBIL optimiser based upon the total number of function evaluations. An improved design strategy employing much less function evaluations is the target for future work.

7. Acknowledgments

The authors are grateful for the support from The Royal Golden Jubilee PhD Program (RGJ-TRF) and the Thailand Research Fund.

8. References

- [1] Kaya, N., Karen, I., Öztürk, F., 2010. Re-design of a failed clutch fork using topology and shape optimisation by the response surface method. *Materials and Design*, Vol. 31, pp. 3008-3014.
- [2] Torstenfelt, B., Klarbring, A., 2007. Conceptual optimal design of modular car product families using simultaneous size, shape and topology optimization, *Finite Elements in Analysis and Design*, Vol. 43, pp. 1050-1061.
- [3] Bureerat, S., Sriworamas, K., 2007. Population-Based Incremental Learning for Multiobjective Optimisation, *Advance in Soft Computing*, Vol. 9, pp. 223-231.
- [4] Kanyakam, S., Bureerat, S., 2007. Passive Vibration Suppression of a Walking Heat and Mass Tractor Handlebar Structure Using Multiobjective PBIL. In: *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore.
- [5] Kanyakam, S., Srisomporn, S., Bureerat, S., 2009. Optimal Geometrical Design of Multiple Heights Pin-Fin Heat Sink Using MOPBIL. In: *Proceedings of the 23rd Conference of the Mechanical Engineering Network of Thailand*, Chiang Mai, Thailand.
- [6] S., Srisomporn, S., Bureerat, 2008. Geometrical Design of Plate-Fin Heat Sinks Using Hybridization of MOEA and RSM, *IEEE TRANSACTIONS ON COMPONENTS AND PACKAGING TECHNOLOGIES*, Vol. 31, pp. 351-360.
- [7] Liao, X., Li, Q., Yang, X., Zhang, W., Li, W., (2008). Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Struct Multidisc Optim*, Vol. 35, pp. 561-569.
- [8] Lophaven, SN., Neilson, HB., Sondergaard, J., DACE a MATLAB Kriging Toolbox. Technical report IMM-TR-2002-12, Technical University of Denmark.
- [9] Zhou, Z., Ong, YS., Lim, MH., 2007. Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Computing*, Vol. 11, pp. 957-971.
- [10] Farina, M., Amato, P., 2005. Linked interpolation-optimization strategies for multicriteria optimization problems. *Soft computing*, Vol. 9, pp. 54-65.
- [11] Bureerat, S., Srisomporn, S., 2010. Optimum plate-fin heat sinks by using a multi-objective evolutionary algorithm. *Engineering Optimization*, Vol. 42, pp. 305-343.