



รายงานวิจัยฉบับสมบูรณ์

โครงการ อัลกอริทึมการสืบค้นที่มีประสิทธิภาพสำหรับสร้างแผนการทดลองที่เหมาะสมเพื่อ
จำลองการทดลองด้วยคอมพิวเตอร์

โดย ผศ.ดร. จรัสศรี รุ่งรัตนอุบล และคณะ

พฤษภาคม 2556

สัญญาเลขที่ R2555B015

อภินันทนาการ

รายงานวิจัยฉบับสมบูรณ์



สำนักหอสมุด

โครงการ อัลกอริทึมการสืบค้นที่มีประสิทธิภาพสำหรับสร้างแผนการทดลองที่เหมาะสมเพื่อ
จำลองการทดลองด้วยคอมพิวเตอร์

- คณะผู้วิจัย
1. ผศ.ดร. จรัสศรี รุ่งรัตนอุบล สังกัด ภาควิชาวิทยาการคอมพิวเตอร์และ
เทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์
ม.นเรศวร
 2. ผศ.ดร. อนามัย นาอุดม สังกัด ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
ม.นเรศวร

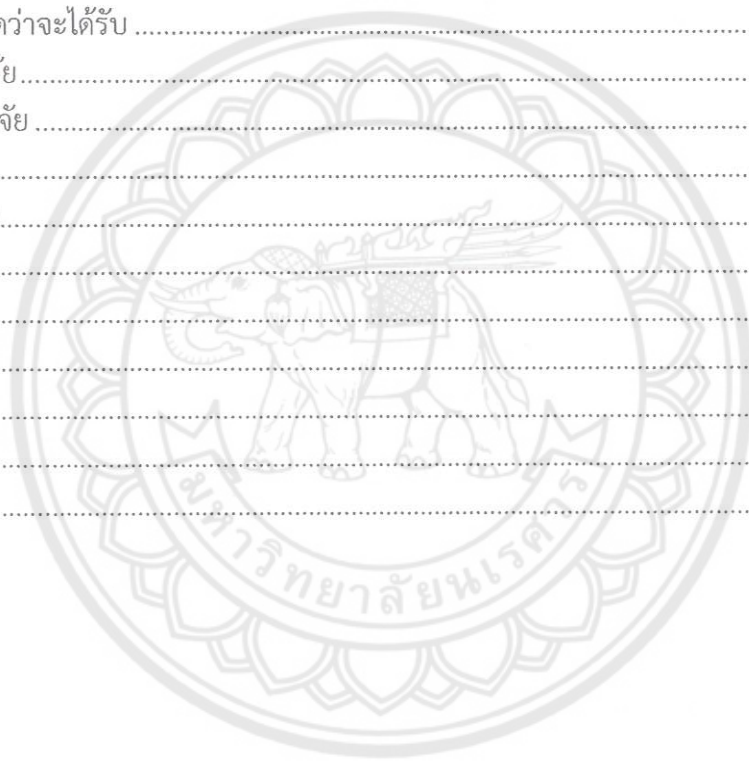
สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน..... 19 ก.ค. 2556
เลขทะเบียน..... 16337405
เลขเรียกหนังสือ..... ๖ ๑A

๙. A๕๖
๗1615
๒556

สนับสนุนโดยงบประมาณแผ่นดิน มหาวิทยาลัยนเรศวร

สารบัญ

บทคัดย่อ	3
Abstract	4
Executive Summary	5
บทที่ 1.....	7
บทนำ.....	7
บทที่ 2.....	11
กรอบแนวคิดทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	11
วัตถุประสงค์ของการวิจัย.....	16
ประโยชน์ที่คาดว่าจะได้รับ	17
ขอบเขตการวิจัย.....	17
วิธีดำเนินการวิจัย	18
บทที่ 3.....	19
วิธีดำเนินการวิจัย.....	19
บทที่ 4.....	30
ผลการวิจัย.....	30
บทที่ 5.....	35
สรุปผลการวิจัย.....	35
บรรณานุกรม	36
ภาคผนวก	39



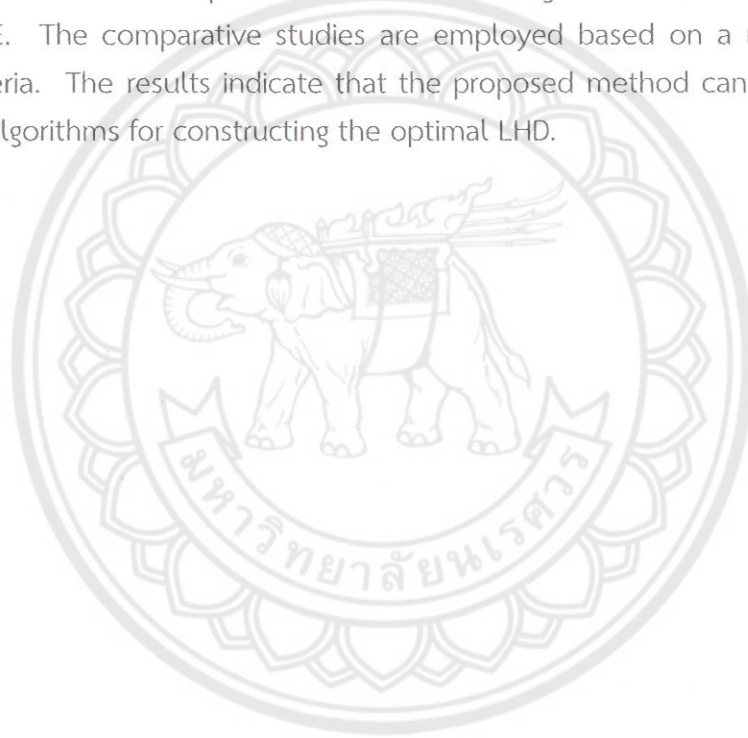
บทคัดย่อ

การจำลองการทดลองด้วยคอมพิวเตอร์ถูกนำมาใช้อย่างแพร่หลายในงานวิจัยด้านวิทยาศาสตร์ และด้านวิศวกรรมศาสตร์ ประสิทธิภาพและผลที่ได้จากการจำลองการทดลองด้วยคอมพิวเตอร์นั้นจะขึ้นอยู่กับการวางแผนทดลองเพื่อเลือกกลุ่มของตัวแปรอิสระเข้ามาทำการทดลองด้วยคอมพิวเตอร์ โดยทั่วไปการค้นหาแผนการทดลองที่เหมาะสมที่สุดสามารถทำได้โดยการใช้อัลกอริทึมการสืบค้นควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม งานวิจัยนี้จึงมุ่งศึกษาเพื่อปรับปรุงประสิทธิภาพของ SA และ ESE ให้ดียิ่งขึ้น ซึ่งผลจากการวิจัยครั้งนี้แสดงว่า SA ที่ปรับปรุงสามารถสืบค้นแผนการทดลองที่เหมาะสมที่สุดของแต่ละมิติปัญหาใดๆ ได้อย่างรวดเร็ว และการปรับปรุงประสิทธิภาพของ ESE โดยใช้หลักการค้นหาของ SA เข้ามามีส่วนประกอบด้วยจะมีประสิทธิภาพมากยิ่งขึ้นและเวลาในการสืบค้นน้อยลงอย่างมากในการสร้างแผนการทดลองแบบ LHD



Abstract

Computer simulated experiments (CSE) are often used in science and engineering applications. The nature of CSE is that time consuming and computationally expensive to run. Normally, the output response from computer simulated experiments is deterministic. Consequently the space filling designs, which focus on spreading design points over a design space, are necessary. Latin hypercube designs (LHD) are normally practiced in the context of CSE. The optimal LHD for a given dimension of problem is constructed by using a search algorithm under a pre-specified optimality criterion. Usually this searching process takes a long time to terminate, especially when the dimension of the problem is large. This paper proposes the methods to enhance the performance of search algorithms which are widely used in the context of CSE. The comparative studies are employed based on a range of problems and optimality criteria. The results indicate that the proposed method can improve the capability of the search algorithms for constructing the optimal LHD.



Executive Summary

ในปัจจุบันการจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiments: CSE) ได้เข้ามามีบทบาทสำคัญในการศึกษารูปแบบความสัมพันธ์ของตัวแปรเข้า (Input variable) และตัวแปรออก (Output response) ที่มีความซับซ้อนและไม่สามารถทำการทดลองทางกายภาพ (Physical experiment) ได้ เช่น การวิเคราะห์หาปริมาณน้ำมันใต้ดินในบางพื้นที่เมื่อทราบคุณลักษณะบางประการของตัวแปรเข้าที่เกี่ยวข้องในพื้นที่นั้นๆ การศึกษาลักษณะการไหลเวียนของโลหิตเมื่อทำการทดลองฉีดยาให้กับผู้ป่วย ซึ่งจะเห็นได้ว่าการทดลองลักษณะนี้ ไม่สามารถทำการทดลองกับหน่วยทดลองจริง ๆ ได้ จึงจำเป็นต้องใช้เทคนิคการจำลองการทดลองด้วยคอมพิวเตอร์หรือ CSE เข้ามาช่วยในการศึกษา โดยทั่วไปแล้วการทำงานของ CSE มักจะใช้เวลานานและสิ้นเปลืองทรัพยากรมาก [10] ดังนั้นนักวิจัยจึงได้ทำการคิดค้นวิธีการหากลุ่มของระดับตัวแปรเข้าที่เหมาะสม หรือ แผนการทดลองที่เหมาะสมเพื่อนำไปใช้ในการจำลองการทดลองด้วยคอมพิวเตอร์ เพื่อเป็นการประหยัดทรัพยากรและสามารถนำผลลัพธ์ที่ได้หรือค่าตัวแปรออกไปใช้อย่างมีประสิทธิภาพ

ลักษณะสำคัญของ CSE คือค่าตัวแปรออกที่ได้จากการจำลอง จะมีลักษณะตรงแบบ (Deterministic) กล่าวคือทุก ๆ ครั้งที่มีการประมวลผลด้วย CSE ที่ระดับค่าคงที่ของกุ่มตัวแปรเข้า จะส่งผลให้ได้ค่าตัวแปรออกที่คงที่เสมอ ดังนั้นแผนการทดลองที่เหมาะสมสำหรับการทดลองประเภทนี้จึงเป็นแผนการทดลองแบบเติมเต็มปริภูมิ (Space filling design) ซึ่งแผนการทดลองที่ถูกนำมาใช้อย่างแพร่หลายมีชื่อเรียกว่า แผนการทดลองแบบลาตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) ในทางปฏิบัติการค้นหาแผนการทดลองแบบ LHD ที่เหมาะสมมักจะยุ่งยากและมีความซับซ้อน ยกตัวอย่างเช่น กรณีการสร้างแผนการทดลองที่ประกอบด้วยตัวแปรเข้าจำนวน d ตัวและจำนวนจุดทดลอง (Design run) เท่ากับ n จุด จะเห็นได้ว่ามีแผนการทดลองแบบ LHD ที่เป็นไปได้ทั้งหมดจำนวน $(n!)^d$ แผนการทดลองซึ่งจำนวนนี้จะเพิ่มขึ้นเรื่อย ๆ เป็นจำนวนมหาศาลเมื่อ n และ d มีขนาดใหญ่ขึ้น ซึ่งจะเห็นได้ว่าการสร้างแผนการทดลองที่เหมาะสมที่สุดจัดเป็นปัญหา NP hard ที่ไม่สามารถทำได้ด้วยมือเปล่า จึงจำเป็นต้องอาศัยการค้นหาแผนการทดลองที่เหมาะสมที่สุดโดยใช้อัลกอริทึมการค้นหา (Search algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria) ซึ่งอัลกอริทึมการค้นหาที่ถูกนำมาใช้ในการด้าน CSE จะใช้เทคนิคการสับเปลี่ยนสมาชิกในแต่ละจุดทดลองเพื่อให้เกิดแผนการทดลองแบบใหม่และพิจารณาเปรียบเทียบแผนการทดลองใหม่ที่เกิดจากการสลับที่ว่ามีคุณสมบัติดีขึ้นหรือไม่ ถ้าพบว่าแผนการทดลองแบบใหม่ที่ดีขึ้นก็จะทำการแทนที่แผนการทดลองเดิมด้วยแผนการทดลองตัวใหม่ที่ได้ ในทางตรงกันข้ามถ้าพบว่าแผนการทดลองตัวใหม่มีคุณสมบัติที่แย่ลง ก็จะต้องทำการตัดสินใจว่าจะมีการแทนที่หรือไม่

งานวิจัยนี้จะมุ่งเน้นศึกษาหาปัจจัยที่มีผลต่อประสิทธิภาพของอัลกอริทึมการหาค่าเหมาะสมแต่ละประเภท จากนั้นจะมีการประยุกต์การวิเคราะห์ความไวเข้ามาใช้ เพื่อศึกษาหาเขตของพารามิเตอร์ที่เหมาะสมสำหรับอัลกอริทึมการหาค่าเหมาะสมแต่ละชนิด โดยผู้วิจัยคาดหวังว่าผลที่ได้จะสามารถนำมาสรุปเป็นตารางสำเร็จรูป ซึ่งผู้ใช้สามารถนำไปใช้ได้อย่างรวดเร็ว จากนั้นผู้วิจัยจะนำเสนอแนวคิดในการปรับปรุงการทำงานของแต่ละอัลกอริทึมการหาค่าเหมาะสม เช่นการปรับโครงสร้างย่านใกล้เคียง การปรับกฎการแทนที่ การปรับกฎการหยุดสืบค้น และการปรับวิธีการคำนวณเกณฑ์การเลือกค่าที่เหมาะสม เป็นต้น เพื่อให้อัลกอริทึมแต่ละประเภทสามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ไม่ว่าจะเป็น ในแง่ของอัตราความเร็วในการลู่เข้าหาค่าที่เหมาะสม (Rate of convergence) รวมไปถึง การรายงานแผนการทดลองที่เหมาะสมที่ยังคุณสมบัติที่ดีเท่าเดิมทุกประการ แต่ใช้เวลาทดลองอย่างมีนัยสำคัญทางสถิติ หลังจากทำการปรับปรุงโครงสร้างการทำงานของอัลกอริทึมการหาค่าเหมาะสมแต่ละประเภทแล้ว ก็จะมีการเปรียบเทียบประสิทธิภาพของอัลกอริทึมประเภทต่างๆ เมื่อกำหนดมิติแผนการทดลองที่แตกต่างกัน เพื่อเป็นแนวทางในการเลือกใช้ อัลกอริทึมที่เหมาะสมสำหรับแต่ละปัญหา

ผลที่ได้จากงานวิจัยพบว่า MESE มีประสิทธิภาพสูงกว่า ESE เมื่อพิจารณาที่จำนวนการแลกเปลี่ยนสมาชิกในปัญหาขนาดเล็ก แต่ในปัญหาขนาดใหญ่ MESE ไม่สามารถสร้างแผนการทดลองที่เหมาะสมได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยกว่า ESE มากนัก สืบเนื่องมาจากกฎการหยุดสืบค้นไม่สามารถทำให้ MESE หยุดทำงานได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยลง ซึ่งจากการศึกษาพบว่า เมื่อปัญหามีขนาดใหญ่แผนการทดลองสุดท้ายที่ได้ในแต่ละรอบการทำงานจะให้ค่า Φ_p ที่น้อยลงเสมอในทุกๆ รอบ สรุปคือ MESE มีประสิทธิภาพสูงกว่า ESE เล็กน้อยเมื่อพิจารณาเปรียบเทียบเวลาในการสร้างแผนการทดลองทั้งกับปัญหาขนาดเล็กและปัญหาขนาดใหญ่ นอกจากนี้ยังพบว่าคุณสมบัติของแผนการทดลองที่วัดโดยใช้เกณฑ์การเลือกค่าเหมาะสม Φ_p โดยเฉลี่ยแล้ว MSA, ESE และ MESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ใกล้เคียงกันในปัญหาขนาดเล็ก แต่กับปัญหาขนาดใหญ่ ESE และ MESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ดีกว่า MSA



บทที่ 1

บทนำ

รายงานนี้เป็นส่วนหนึ่งของโครงการวิจัยภายใต้การสนับสนุนโดยงบประมาณแผ่นดิน มหาวิทยาลัยนเรศวร ประจำปีงบประมาณ 2555 โดยคณะผู้วิจัยได้รับการอนุมัติให้ทำงานวิจัยนี้ โดยมีชื่อโครงการและรายละเอียดเกี่ยวกับโครงการวิจัยดังต่อไปนี้

ชื่อโครงการ (ภาษาไทย) อัลกอริทึมการสืบค้นที่มีประสิทธิภาพสำหรับสร้างแผนการทดลองที่เหมาะสมเพื่อจำลองการทดลองด้วยคอมพิวเตอร์

(ภาษาอังกฤษ) The efficient search algorithms for constructing optimal designs in computer simulated experiments

คณะผู้วิจัย(ระดับสังกัดภาควิชา) และสัดส่วนที่ทำงานวิจัย (%)
หัวหน้าโครงการวิจัย

ผศ. ดร. จรัสศรี รุ่งรัตนอุบล (50%)

อาจารย์สังกัดภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

ผู้วิจัยร่วม

ผศ.ดร. อนามัย นาอุดม (50%)

อาจารย์ สังกัดภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์
มหาวิทยาลัยนเรศวร

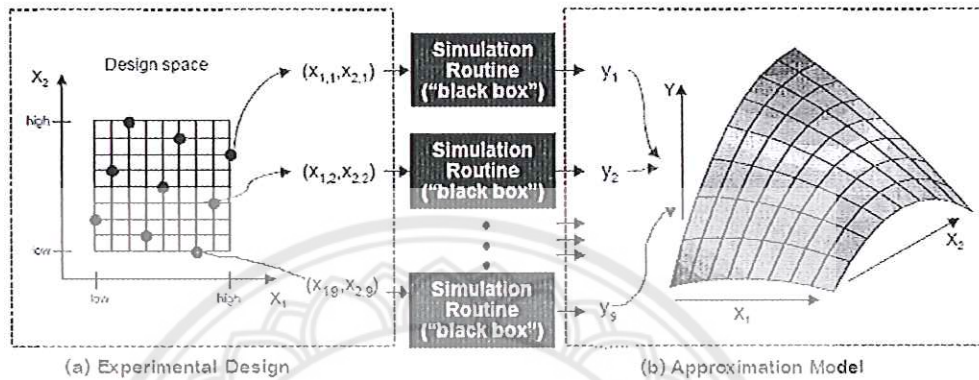
สถานที่จัดทำโครงการวิจัย

ภาควิชาคณิตศาสตร์และภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

ในปัจจุบัน ความเจริญก้าวหน้าด้านเทคโนโลยีเป็นไปอย่างรวดเร็ว ส่งผลให้มนุษย์มีการประยุกต์ใช้เทคโนโลยีด้านคอมพิวเตอร์และคณิตศาสตร์ประยุกต์ที่มีชื่อเรียกว่า เทคนิคการจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiments: CSE) อย่างแพร่หลาย เพื่อทำการศึกษารูปแบบความสัมพันธ์ระหว่างกลุ่มตัวแปรเข้า (Input variables) และตัวแปรออก (Output response) ของระบบที่มีความซับซ้อนและไม่สามารถทำการทดลองทางกายภาพ (Physical experiment) ทั่วไปได้ [10, 31, 36] ยกตัวอย่างเช่น การจำลองตัวแบบ 3 มิติทางคณิตศาสตร์ผ่านตัวจำลองคอมพิวเตอร์เพื่อวิเคราะห์หาปริมาณน้ำมัน

ได้คืนในบางพื้นที่เมื่อทราบคุณลักษณะบางประการของตัวแปรเข้าที่เกี่ยวข้องในพื้นที่นั้น ๆ [18] หรือการใช้การคำนวณพลศาสตร์ของไหล (Fluid dynamics method) เพื่อศึกษาลักษณะการไหลเวียนของโลหิตเมื่อทำการทดลองฉีดยาให้กับผู้ป่วย เป็นต้น จะเห็นได้ว่าการทดลองลักษณะนี้ ไม่สามารถทำการทดลองกับหน่วยทดลองจริง ๆ ได้ เนื่องจากข้อจำกัดด้านค่าใช้จ่ายและความปลอดภัยของหน่วยทดลอง จึงต้องหลีกเลี่ยงการวางแผนทดลองทางกายภาพโดยการใช้ CSE เข้ามาแทนที่เพื่อช่วยในการศึกษาคุณลักษณะของระบบที่สนใจนั่นเอง กระบวนการทำงานของการจำลองการทดลองด้วยคอมพิวเตอร์หรือ CSE สามารถอธิบายได้ด้วยรูปที่ 1 ดังนี้



ภาพ 1 แสดงการทำงานของ การจำลองการทดลองด้วยคอมพิวเตอร์ (CSE) ดัดแปลงจาก [33]

จากรูปที่ 1 จะเห็นได้ว่ากระบวนการทำงานใน CSE สามารถแบ่งได้เป็น 3 ขั้นตอนคือ

1. Experimental design คือ การวางแผนการทดลอง ซึ่งประกอบด้วยการจัดค่าระดับต่าง ๆ (Treatment combination) ของกลุ่มตัวแปรเข้าที่เกี่ยวข้องทั้งหมดเพื่อทำการจำลอง
2. Simulation routine หรือ กล่องดำ (Black box) คือ กระบวนการทำงานภายในระบบที่เราไม่สามารถมองเห็นและคาดการณ์ได้ว่าจะเกิดอะไรขึ้นภายใน ซึ่งกระบวนการในกล่องดำนี้ จะใช้รูปแบบสมการทางคณิตศาสตร์ที่มีความซับซ้อน ประมวลผลเพื่อให้ได้ผลลัพธ์คือค่าตัวแปรออก และค่าผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ ของตัวแปรเข้าที่เราป้อนเข้าไป
3. Approximation model คือ การสร้างตัวแบบเพื่อการพยากรณ์ โดยการนำค่าของตัวแปรเข้าที่ทำการทดลองและค่าตัวแปรออกที่ได้จากกล่องดำจำนวนหนึ่ง มาสร้างรูปแบบความสัมพันธ์ที่เหมาะสม และนำไปใช้ประโยชน์ในการพยากรณ์ค่าของตัวแปรออก เมื่อทราบค่าของกลุ่มตัวแปรเข้า

สมมติให้ $X \in R^d$ (d แทนจำนวนตัวแปรเข้าที่สนใจศึกษา) เป็นเวกเตอร์ของระดับค่าต่าง ๆ ของตัวแปรเข้าที่จะทำการประมวล CSE ดังนั้นกระบวนการของ CSE สามารถเขียนแทนด้วยสมการทางคณิตศาสตร์ดังนี้

$$y = f(x) \quad (5.1)$$

จากสมการที่ (5.1) จะได้ว่าค่าของตัวแปรออกที่ได้จากกระบวนการ CSE นั้นจะขึ้นอยู่กับค่าระดับของตัวแปรเข้าที่ป้อนเข้าในระบบ ซึ่งโดยทั่วไปแล้วฟังก์ชัน f จะเป็นค่าที่เราไม่ทราบซึ่งแทนด้วยกล่องดำ ลักษณะสำคัญของ CSE คือ แต่ละหน่วยทดลองหรือรัน (n) จะใช้เวลาและสิ้นเปลืองทรัพยากรมาก ดังนั้นนักวิจัยส่วนใหญ่จึงพยายามพัฒนาตัวแบบทางสถิติ (Statistical modeling methods) โดยใช้จำนวนหน่วยทดลองให้น้อยที่สุด แต่ให้ได้ข้อมูลที่สามารถอธิบายรูปแบบความสัมพันธ์ของตัวแปรเข้าและตัวแปรออกได้มากที่สุด หรือการหา \hat{f} ที่ใกล้เคียง f ให้มากที่สุด ซึ่งหมายถึงการพัฒนาตัวแบบทางสถิติหรือการพัฒนา Surrogate model [1, 31, 34] นั่นเอง ดังนั้นจึงสามารถกล่าวได้ว่างานวิจัยส่วนใหญ่ที่ถูกพัฒนาขึ้นมาในงานด้าน CSE จะมีความเกี่ยวข้องกับการวางแผนการทดลอง (Experimental design) และการสร้างตัวแบบทางสถิติเพื่อการพยากรณ์ (Statistical

approximation model) ซึ่งโครงการวิจัยนี้จะเน้นในส่วนของการสร้างแผนการทดลองที่เหมาะสมสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์

โดยทั่วไปการศึกษาด้าน CSE จะใช้แผนการทดลองแบบละตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) [23] ควบคู่กับตัวแบบทางสถิติ Kriging model [7, 25, 31, 36] ซึ่งมีผลรายงานจากงานวิจัย ระบุว่า การเลือกแผนการทดลองที่เหมาะสม (Optimal design) มีบทบาทสำคัญมากต่อความแม่นยำ (Accuracy) และความน่าเชื่อถือ (Reliability) ของตัวแบบทางสถิติที่จะพัฒนาขึ้นมาเพื่อเป็นตัวแทนของระบบที่มีความซับซ้อน [1, 6, 11, 18, 33, 34] และสามารถนำตัวแบบที่สร้างขึ้นไปพยากรณ์ค่าตัวแปรออกได้อย่างแม่นยำเมื่อทราบค่าของตัวแปรเข้าใด ๆ อย่างไรก็ตาม มีการรายงานไว้ในงานวิจัยหลายชิ้น [5, 15, 19, 21, 25, 36, 38] ที่ระบุว่า การสร้างแผนการทดลองแบบ LHD ที่เหมาะสมที่สุดหรือ Optimal Latin hypercube design (OLHD) นั้นไม่ใช่เรื่องง่าย เพราะเป็นปัญหาประเภท NP-hard ซึ่งต้องใช้เวลานานมากในการค้นหา [15, 19]

การสร้างแผนการทดลองแบบ LHD ที่เหมาะสมหรือ OLHD นั้น สามารถทำได้โดยการพิจารณาคุณสมบัติของแผนการทดลองที่สร้างขึ้นว่ามีคุณสมบัติที่ดีหรือไม่โดยพิจารณาจากความเต็มเต็มปริภูมิของการทดลอง (Space filling) เพื่อให้มั่นใจได้ว่าแผนการทดลองที่สร้างขึ้นครอบคลุมขอบเขตของการศึกษาได้ครบถ้วน ซึ่งคุณสมบัตินี้สามารถพิจารณาจากค่าเกณฑ์การเลือกค่าเหมาะสม (Optimality criteria) เช่น Integrated mean square error (IMSE) [31] เกณฑ์ Entropy [18] เกณฑ์ Maximin distance [16] และเกณฑ์ ϕ_p [25] เป็นต้น และคุณสมบัติของแผนการทดลองที่สำคัญอีกประการหนึ่งคือ คุณสมบัติเชิงตั้งฉาก (Orthogonality) ซึ่งเป็นคุณสมบัติที่สำคัญมากเพื่อให้มั่นใจได้ว่าการประมาณอิทธิพลของตัวแปรเข้าต่าง ๆ ที่มีต่อตัวแปรออกเป็นไปอย่างอิสระกัน ซึ่งเกณฑ์ที่นิยมใช้คือ Minimum pair-wise correlation [17, 29, 35] จากการรวบรวมงานวิจัยที่เกี่ยวข้องเบื้องต้นพบว่า เราสามารถแบ่งวิธีการสร้างแผนการทดลองออกเป็น 2 วิธีที่แตกต่างกัน วิธีแรกเป็นการสร้างโดยใช้หลักการทางพีชคณิต (Algebraic method) หรือวิธี Non search-based ตามที่ได้เสนอไว้ในงานวิจัยหลายชิ้น [3, 11, 37] ผลที่ได้จากงานวิจัยเหล่านี้พบว่าแผนการทดลองที่สร้างจากหลักการนี้มีค่าเกณฑ์เลือกค่าเหมาะสมที่ดีและสามารถนำไปสร้างตัวแบบทางสถิติที่ความแม่นยำในการพยากรณ์แต่มีข้อจำกัดคือสามารถสร้างแผนการทดลองได้บางมิติเท่านั้น เช่นในงานของ Butler [3] ที่เสนอการวางแผนทดลองโดยประยุกต์ใช้ William transformation ซึ่งจะต้องกำหนดจำนวนรันเป็นเลขเฉพาะเท่านั้น นอกจากนี้งานของ Ye [38] ที่ใช้วิธีการของ Kronecker product มาสร้างแผนการทดลองที่เหมาะสม ซึ่งมีประสิทธิภาพดีแต่มีข้อจำกัดคือ $n = 2^m + 1$ และ $d = 2m - 2$ เท่านั้น วิธีที่ 2 ของการสร้างแผนการทดลองแบบ LHD ที่เหมาะสม หรือ OLHD คือการใช้ อัลกอริทึมการค้นหา (Search algorithm) ควบคู่กับการพิจารณาเกณฑ์เลือกค่าเหมาะสม (Optimality criteria) ซึ่งหลักการโดยทั่วไปของวิธีการนี้คือการใช้เทคนิคการสลับสมาชิกจุดทดลองคู่ใด ๆ เพื่อให้เกิดการกระจายจุดทดลองแบบเต็มเต็มปริภูมิการทดลองโดยพิจารณาจากค่าเกณฑ์เลือกค่าเหมาะสมว่าดีขึ้นหรือไม่ โดยอัลกอริทึมการค้นหาจะหยุดการทำงานเมื่อความแตกต่างระหว่างค่าเกณฑ์เลือกค่าเหมาะสมใน Iteration ปัจจุบัน และค่าใน Iteration ก่อนหน้าไม่มีความแตกต่างกันมากพอ (พิจารณาจากค่า Tolerance level ที่กำหนดไว้ล่วงหน้า) ซึ่งเป็นที่แน่นอนว่าบางครั้งกระบวนการสลับจะสิ้นสุดการสลับ ณ ตำแหน่งที่ห่างจากค่า Global optimum มาก ๆ ก็ได้ อัลกอริทึมการค้นหาหลายประเภทถูกนำเสนอขึ้นมาเพื่อใช้ค้นหาแผนการทดลอง OLHD เช่น Columnwise-pairwise algorithm (CP) [20], Simulated annealing algorithm (SA) [25] และ Genetic algorithm (GA) [21], Evolutionary search algorithm [15] เป็นต้น และเกณฑ์ในการเลือกค่าที่เหมาะสมหลายแบบถูกเสนอขึ้นมาเพื่อใช้ในการตัดสินใจว่าแผนการทดลองที่ถูกค้นพบนั้นมีคุณสมบัติที่ดีพอหรือยัง สำหรับเกณฑ์ที่นิยมใช้อย่างแพร่หลายจะเป็นการวัดระยะห่างระหว่างจุดทดลองใด ๆ ภายในปริภูมิของการทดลอง เช่น ระยะทางแบบยูคลิด (Euclidean distance) แต่ข้อจำกัดในการใช้อัลกอริทึมในการสืบค้นควบคู่กับเกณฑ์เลือก

ค่าที่เหมาะสมคือกระบวนการสืบค้นมักใช้เวลายาวนานมาก โดยเฉพาะอย่างยิ่งเมื่อมิติของแผนการทดลองมีขนาดใหญ่ขึ้น

จากที่กล่าวมาทั้งหมด สามารถสรุปได้ว่าการสร้างแผนการทดลอง OLHD สำหรับ CSE โดยใช้หลักการทางพีชคณิตมีข้อดีคือไม่จำเป็นต้องใช้เวลาในการสืบค้นแต่ขาดความยืดหยุ่นในแง่ของการกำหนดมิติปัญหาที่สนใจศึกษา ส่วนข้อดีของการสร้างแผนการทดลองโดยใช้อัลกอริทึมการสืบค้นคือ ถึงแม้ว่าจะต้องใช้เวลาในการกระบวนการสืบค้น แต่ค่อนข้างจะยืดหยุ่นสำหรับทุกมิติปัญหาที่สนใจศึกษา ดังนั้นจึงมีการพัฒนาแนวคิดในการสร้างแผนการทดลองแบบ OLHD โดยการผนวกวิธีการทางพีชคณิตเข้ากับอัลกอริทึมการสืบค้นเพื่อให้สามารถสร้างแผนการทดลองได้อย่างมีประสิทธิภาพมากขึ้น ดังจะเห็นได้จากกรายงานวิจัยหลายชิ้น [12, 19, 29, 38] ทั้งนี้วิธีการดังกล่าวยังมีความจำเป็นต้องขยายขอบเขตของปัญหาออกไปอีกไม่ว่าจะเป็นด้านการเพิ่มขนาดของมิติปัญหา รวมไปถึงการประยุกต์แนวคิดทางพีชคณิตแบบอื่น ๆ เช่น กรุปวัฏจักร (Cyclic group) เข้ามาประยุกต์ใช้ในการสร้าง OLHD นอกจากนี้ผู้วิจัยยังพบว่าการศึกษาที่ผ่านมาเน้นศึกษาเกณฑ์เลือกค่าเหมาะสมเพียงเกณฑ์ใดเกณฑ์หนึ่งเท่านั้นและก่อให้เกิดคำถามตามมาว่าเกณฑ์เลือกค่าเหมาะสมที่แตกต่างกันก่อให้เกิดแผนการทดลองที่ดีแตกต่างกันหรือไม่ในแง่ของการนำไปใช้สร้างตัวแบบทางสถิติเพื่อการพยากรณ์ นอกจากนี้แนวทางการแก้ปัญหาด้านเวลาที่ใช้ในกระบวนการค้นหายังคงเป็นปัญหาเปิดที่ต้องมีพัฒนาให้ดีขึ้นเรื่อยๆ โดยการหาทางปรับปรุงขั้นตอนการทำงานของอัลกอริทึมแต่ละประเภท เช่น การปรับแผนการทดลองตั้งต้น การปรับกฎการแทนที่ (Replacement rule) การปรับกฎการหยุดสืบค้น (Stopping rule) เป็นต้น ผู้วิจัยคาดว่าผลที่ได้จากการวิจัยนี้จะส่งผลดีต่อการพัฒนาการค้นหายังหรือการสร้างแผนการทดลองที่เหมาะสมโดยใช้การผสมผสานระหว่างวิธีการทางพีชคณิตและอัลกอริทึมการสืบค้นที่มีประสิทธิภาพ โดยเฉพาะอย่างยิ่งในกรณีที่มีมิติของปัญหาที่มีขนาดใหญ่ขึ้น



บทที่ 2

กรอบแนวคิดทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับความสำคัญและที่มาของปัญหาวิจัย รวมไปถึงการทบทวนวรรณกรรมของการศึกษาที่เกี่ยวข้อง และรายละเอียดโดยรวมของอัลกอริทึมการสืบค้นสำหรับการค้นหาแผนการทดลองที่เหมาะสมบนคลาสของการออกแบบการทดลองแบบละตินไฮเปอร์คิวบ์ เพื่อนำไปใช้ในการจำลองการทดลองด้วยคอมพิวเตอร์ ซึ่งรายละเอียดเป็นดังนี้

กรอบแนวคิดในการทำวิจัย

การสร้างแผนการทดลองที่เหมาะสมสำหรับงานด้าน CSE นั้นจะอาศัยวิธีฮิวริสติก (Heuristic method) ซึ่งเป็นวิธีการหาคำตอบที่เหมาะสมจากกลุ่มคำตอบที่เป็นไปได้ที่สุ่มมาทั้งหมด ซึ่งจะเกี่ยวข้องกับปัญหาการหาค่าเหมาะสม (Optimization problem) กล่าวคือ ถ้ากำหนดให้ $X \in R^d$ ซึ่ง R^d เป็นปริภูมิแบบจำกัดแต่มีมิติขนาดใหญ่ และกำหนดให้ $h(x)$ แทนฟังก์ชันวัตถุประสงค์ของ $X \in R^d$ ใด ๆ ดังนั้นปัญหาการหาค่าเหมาะสมในการสร้างแผนการทดลองแบบเต็มเต็มปริภูมิ คือการหาแผนการทดลอง X^* ใดๆ ที่สอดคล้องกับเงื่อนไขต่อไปนี้

$$h(X^*) = \min_{X \in R^d} h(X) \quad (1)$$

ฟังก์ชันวัตถุประสงค์ $h(\cdot)$ อาจหมายถึงเกณฑ์เลือกค่าเหมาะสมใด ๆ ซึ่งจะต้องมีการกำหนดไว้ล่วงหน้าก่อนจะทำการค้นหา

ตามที่ได้กล่าวไว้ข้างต้นแล้วว่า อัลกอริทึมการหาค่าเหมาะสมที่ถูกนำมาใช้ในงานด้าน CSE อย่างแพร่หลาย ประกอบไปด้วย CP, SA และ GA ซึ่งอัลกอริทึมเหล่านี้มีขั้นตอนการทำงานที่คล้ายคลึงกัน และสามารถสรุปหลักการการทำงานของอัลกอริทึมเหล่านี้ ได้ดังต่อไปนี้

ขั้นตอนที่ 1: เริ่มต้นการค้นหาโดยการสุ่มเลือกแผนการทดลองตั้งต้น X^0 และกำหนดให้ $X^C = X^0$

ขั้นตอนที่ 2: สร้างโครงสร้างย่านใกล้เคียง (Neighborhood structure) ของ X^C ตามนิยามของโครงสร้างที่ได้ระบุไว้ล่วงหน้า จากนั้นเลือกแผนการทดลอง X^{new} จากแผนการทดลองที่เป็นไปได้ทั้งหมดของโครงสร้างย่านใกล้เคียงที่สร้างขึ้นมา

ขั้นตอนที่ 3: คำนวณฟังก์ชันวัตถุประสงค์ของแผนการทดลอง X^{new} ที่เลือกมา และตัดสินใจว่าจะทำการแทนที่ X^C ด้วย X^{new} หรือไม่ จากนั้นย้อนกลับไปทำขั้นตอนที่ 2 ซ้ำไปเรื่อย ๆ เพื่อค้นหาแผนการทดลองที่มีคุณสมบัติดีขึ้นเรื่อย ๆ จนกว่าจะบรรลุเงื่อนไขของกฎการหยุดสืบค้น (Stopping rule) และทำการรายงานแผนการทดลองที่ได้ ซึ่งจะถือว่าเป็นแผนการทดลองที่เหมาะสมที่สุด ที่ได้จากกระบวนการสืบค้น

จากขั้นตอนการทำงานของอัลกอริทึมหาค่าเหมาะสมที่กล่าวมาข้างต้น รวมไปถึงข้อจำกัดด้านเวลา ในกรณีที่ต้องการวางแผนการทดลองที่เหมาะสม สำหรับปัญหาที่มีมิติขนาดใหญ่ขึ้น ผู้วิจัยจึงสนใจศึกษา วิธีการปรับปรุงประสิทธิภาพของอัลกอริทึมเหล่านี้เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น ซึ่งต่อไปนี้จะได้อธิบายถึงองค์ประกอบต่าง ๆ ที่มีอิทธิพลต่อประสิทธิภาพการทำงานของอัลกอริทึมต่าง ๆ รวมไปถึงไปแนวทางในการปรับปรุงเพื่อให้อัลกอริทึมสามารถทำงานได้ดียิ่งขึ้น รายละเอียดต่าง ๆ เป็นดังต่อไปนี้

1. การกำหนดจุดตั้งต้น (Starting point) จากขั้นตอนที่ 1 ของการทำงานของอัลกอริทึมการหาค่าเหมาะสม ซึ่งเป็นการกำหนดแผนการทดลองตั้งต้น ซึ่งจะเห็นได้ว่าเป็นการกำหนดแบบสุ่มขึ้นมา (Random) ผู้วิจัยมีแนวคิดที่ว่า แทนที่เราจะเริ่มค้นหาแบบสุ่ม เราสามารถประยุกต์ใช้ทฤษฎีการวางแผนทดลอง เช่น กฎของการตั้งฉากระหว่างคอลัมน์ในแผนการทดลอง (Orthogonal array) หรือ กฎการสมมาตร (Symmetry law) ของจุดทดลอง โดยมีการกำหนดจุดให้กระจายอยู่ในบริเวณของการทดลองเป็นแบบสมมาตร เมื่อมีการประยุกต์กฎเหล่านี้ เข้ามาช่วยในการกำหนดแผนการทดลองตั้งต้น ก็จะส่งผลให้ขอบเขตของการค้นหามีขนาดลดลง และทำให้การค้นหาสำหรับมิติขนาดใหญ่สามารถทำได้ทั่วถึงอย่างรวดเร็วยิ่งขึ้น นั่นคือใช้เวลาในการค้นหาน้อยลงนั่นเอง

2. โครงสร้างย่านใกล้เคียง (Neighborhood structure) ตามที่ได้ระบุไว้ในขั้นตอนที่ 2 เกี่ยวกับโครงสร้างย่านใกล้เคียง ซึ่งในการวางแผนการทดลองนั้นย่านใกล้เคียงหมายถึง แผนการทดลองที่มีโครงสร้างหรือคุณลักษณะใกล้เคียงกัน ในการกำหนดโครงสร้างของย่านใกล้เคียงสามารถทำได้หลายวิธี แต่วิธีที่นิยมใช้อย่างแพร่หลายในงานด้าน CSE คือการสลับสมาชิกสองแถวใด ๆ ในคอลัมน์ที่ถูกเลือกมาโดยสุ่ม เนื่องจากการสลับแบบนี้จะไม่ส่งผลกระทบต่อ ต่อโครงสร้างของแผนการทดลองแบบ LHD ให้เปลี่ยนแปลงไป ยกตัวอย่างเช่น โครงสร้างย่านใกล้เคียงของแผนการทดลองแบบ LHD ที่มีขนาด 5 แถว และ 4 คอลัมน์ เมื่อมีการสลับสมาชิกระหว่างแถวที่ 2 และแถวที่ 5 ภายในคอลัมน์ที่ 2 ที่ถูกสุ่มเลือกมา จะได้ผลดังรูป

$$\begin{pmatrix} 1 & 2 & 4 & 5 \\ 3 & \boxed{4} & 5 & 3 \\ 2 & 1 & 3 & 4 \\ 4 & \boxed{5} & 1 & 2 \\ 5 & 3 & 2 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 2 & 4 & 5 \\ 3 & \boxed{5} & 5 & 3 \\ 2 & 1 & 3 & 4 \\ 4 & \boxed{4} & 1 & 2 \\ 5 & 3 & 2 & 1 \end{pmatrix}$$

จากรูปดังกล่าว จะเห็นได้ว่ามีจำนวนแผนการทดลองที่เป็นไปได้ทั้งหมดในโครงสร้างย่านใกล้เคียงเท่ากับ $4 \cdot \binom{5}{2} = 40$ แผนการทดลอง ซึ่งเป็นที่ชัดเจนว่าถ้ามิติของแผนการทดลองมีขนาดใหญ่ขึ้น ย่อมส่งผลให้จำนวน

แผนการทดลองที่เป็นไปได้ทั้งหมดในโครงสร้างย่านใกล้เคียงเพิ่มขึ้นตามไปด้วย จะเห็นได้ว่าการเลือกใช้โครงสร้างย่านใกล้เคียงที่เหมาะสมจึงเป็นสิ่งสำคัญอย่างยิ่ง จากแนวคิดเกี่ยวกับโครงสร้างย่านใกล้เคียงนี้ ผู้วิจัยมีจุดมุ่งหมายที่จะปรับปรุงโครงสร้างให้ดียิ่งขึ้นโดยการปรับเปลี่ยนวิธีการสุ่มโดยจะทำการสุ่มที่หลายคอลัมน์เพื่อทำการสลับสมาชิกในสองแถวใด ๆ แทนที่จะสุ่มเพียงคอลัมน์เดียวเท่านั้น ซึ่งจากการศึกษาค้นคว้าวิจัยเบื้องต้น ผู้วิจัยพบว่าวิธีการใหม่นี้ จะทำให้การค้นหาแผนการทดลองเหมาะสมสามารถลู่อู่เข้าหาแผนการทดลองที่มีคุณสมบัติดีขึ้น ได้อย่างรวดเร็วมากขึ้น

3. กฎการแทนที่ (Replacement rule) นอกจากปัจจัยที่ได้กล่าวมาแล้ว ยังมีองค์ประกอบที่สำคัญต่อประสิทธิภาพการค้นหาแผนการทดลองเหมาะสมของอัลกอริทึมการหาค่าเหมาะสม คือกฎการแทนที่ (Replacement rule) ในขั้นตอนที่ 3 ของกระบวนการสืบค้น เมื่อเลือกแผนการทดลองจากโครงสร้างย่านใกล้เคียงแล้วจะต้องมีการตัดสินใจว่าจะมีการแทนที่ X^C ด้วย X^{new} หรือไม่ ยกตัวอย่างเช่น กฎการแทนที่ใน CP นั้น สามารถทำได้โดยการกำหนดให้ $\nabla h = h(X^{new}) - h(X^C)$ และจะได้ว่า

$$\text{CP rule} = \begin{cases} 1, & \text{if } \nabla h \leq 0; \\ 0, & \text{if } \nabla h > 0. \end{cases} \quad (2)$$

เมื่อ “1” หมายถึงมีการแทนที่ และ “0” หมายถึงไม่มีการแทนที่ จะเห็นได้ว่ากฎการแทนที่ของ CP ค่อนข้างง่ายและไม่ซับซ้อน อย่างไรก็ตาม จะส่งผลให้ CP หยุดการค้นหาที่ค่าเหมาะสมเฉพาะที่ (Local optimum) ได้ ซึ่งแนวทางการแก้ไขปัญหานี้คือ ทำการค้นหาซ้ำด้วยจำนวนครั้งที่มากพอ และเลือกคำตอบที่ดีที่สุดมา ซึ่งไม่เหมาะสมในทางปฏิบัติ เนื่องจากการทำซ้ำจะทำให้เสียเวลามาก และไม่สามารถทำได้ในกรณีที่มีมิติของการทดลองมีขนาดใหญ่

กฎการแทนที่ใน SA ค่อนข้างจะแตกต่างจาก CP โดยกำหนดว่า จะมีการแทนที่ X^C ด้วย X^{new} ถ้า

$$\exp\left(-\frac{\nabla h}{t}\right) \geq u$$

เมื่อ u แทนเลขสุ่มที่ถูกสุ่มมาจากการแจกแจงแบบยูนิฟอร์ม $u \sim U[0,1]$ และ “ t ” เป็นพารามิเตอร์ ที่มีชื่อเรียกว่า “อุณหภูมิ” ซึ่งค่า “ t ” จะถูกทำให้มีค่าลดลงเรื่อย ๆ โดยกำหนดให้ $t = t_0$ ในขั้นตอนเริ่มต้นการค้นหา และกำหนดให้ “ t ” มีอัตราการลดลงเท่ากับ $t = \alpha t_0$ เมื่อ $0 < \alpha < 1$ ซึ่งแทนปัจจัยในการลดอุณหภูมิในกระบวนการค้นหา โดยปกติแล้วถ้าค่า t และ α มีค่ามากเกินไป จะส่งผลให้อัตราเร่งของการลู่เข้าสู่ค่าเหมาะสมเป็นไปอย่างช้า [11] และจากกฎการแทนที่ใน SA จะเห็นได้ว่าแผนการทดลอง X^{new} ที่มีคุณสมบัติแย่งเพียงเล็กน้อยจะมีโอกาสแทนที่ X^C มากกว่าแผนการทดลอง X^{new} ที่มีคุณสมบัติแย่งมาก ๆ จากรายงานวิจัยที่เกี่ยวข้อง [25] พบว่าประสิทธิภาพการทำงานของ SA จะขึ้นอยู่กับ การเลือกพารามิเตอร์ t_0 , ∇h และ α อย่างมาก โดยสรุปแล้วกฎการแทนที่ใน SA คือ

$$\text{SA rule} = \begin{cases} 1, & \text{if } \exp(-\nabla h/t) \geq u; \\ 0, & \text{if } \exp(-\nabla h/t) < u. \end{cases} \quad (3)$$

สำหรับกรณีของอัลกอริทึม GA นั้นมีการใช้กฎการแทนที่ ค่อนข้างจะแตกต่างจาก CP และ SA โดย GA จะทำการพิจารณาค่าคุณสมบัติของทุกแผนการทดลองที่เป็นไปได้ในโครงสร้างย่านใกล้เคียง จากนั้น จะทำการคำนวณค่าความน่าจะเป็น (Probability) ในการถูกเลือก โดยที่แผนการทดลองที่มีคุณสมบัติดีกว่าจะมีโอกาสหรือความน่าจะเป็นในการถูกเลือกสูงกว่านั้นเอง

จากการทบทวนวรรณกรรมของงานวิจัยที่เกี่ยวข้องพบว่า กฎการแทนที่นั้น มีผลต่อประสิทธิภาพการทำงานของแต่ละอัลกอริทึมการหาค่าเหมาะสมอย่างยิ่ง โดยเฉพาะในด้านเวลาของการค้นหาแผนการทดลองที่เหมาะสม ดังนั้นในงานวิจัยนี้ ผู้วิจัยจึงมีแนวคิดในการปรับเปลี่ยนกลไกภายในของกฎการแทนที่ของอัลกอริทึมทั้ง 3 ชนิดตามที่กล่าวไว้ก่อนหน้านี โดยใช้หลักการจำลองข้อมูลด้วยมิติปัญหาที่หลากหลาย จากนั้นทำการวิเคราะห์ทางสถิติ เพื่อหากฎการแทนที่ที่มีความแกร่ง (Robustness) และสามารถนำไปใช้กับอัลกอริทึมทั้ง 3 ชนิดข้างต้น นอกจากนี้ผู้วิจัยมีแนวคิดที่พัฒนางานกฎการแทนที่โดยใช้หลักการปรับเปลี่ยนตัวเองโดยอัตโนมัติ โดยกฎการแทนที่จะเข้มงวดมากขึ้นเมื่อกระบวนการค้นหาเคลื่อนเข้าไปใกล้แผนการทดลองที่เหมาะสม แต่จะมีความยืดหยุ่นมากขึ้นเมื่อกระบวนการสืบค้นเริ่มเคลื่อนที่ห่างจากค่าเหมาะสม

4. กฎการหยุดสืบค้น (Stopping rule) ปัจจัยที่สำคัญต่อการทำงานของอัลกอริทึมการหาค่าเหมาะสมอีกประการหนึ่ง คือกฎการหยุดสืบค้น (Stopping rule) ซึ่งเป็นเกณฑ์ที่ใช้ในการระบุว่ากระบวนการค้นหาแผนการทดลองที่เหมาะสมจะทำการหยุดการค้นหาเมื่อใด เมื่อไหร่ก็ตามที่พบว่าไม่สามารถค้นหาแผนการทดลองที่ดีกว่า X^C ได้อีกแล้ว กระบวนการค้นหาก็จะสิ้นสุดการค้นหาและรายงานแผนการทดลองล่าสุดออกมา ซึ่งจากงานวิจัยที่

ผ่านมา ผู้วิจัยพบว่า การกำหนดค่าที่แตกต่างกันของกฎการหยุดสืบทอด หรือค่า ระดับความคลาดเคลื่อนที่ยอมรับ (Tolerance level) นั้น จะส่งผลกระทบต่ออย่างมากต่อเวลาที่ใช้ในการค้นหา รวมไปถึงคุณสมบัติของแผนการทดลองที่ถูกรายงานเมื่อสิ้นสุดการสืบค้นด้วย จึงเป็นที่น่าสนใจอย่างยิ่งที่จะศึกษาหาระดับที่พอเหมาะของระดับความคลาดเคลื่อนที่ยอมรับ ที่จะทำได้แผนการทดลองที่เหมาะสมภายใต้เวลาที่จำกัด นั้นเอง ซึ่งงานวิจัยนี้จะใช้หลักการวิเคราะห์ความไว (Sensitivity analysis) เพื่อหาค่ากฎการหยุดสืบทอดที่เหมาะสมสำหรับแต่ละขนาดของมิติปัญหาที่สนใจศึกษา

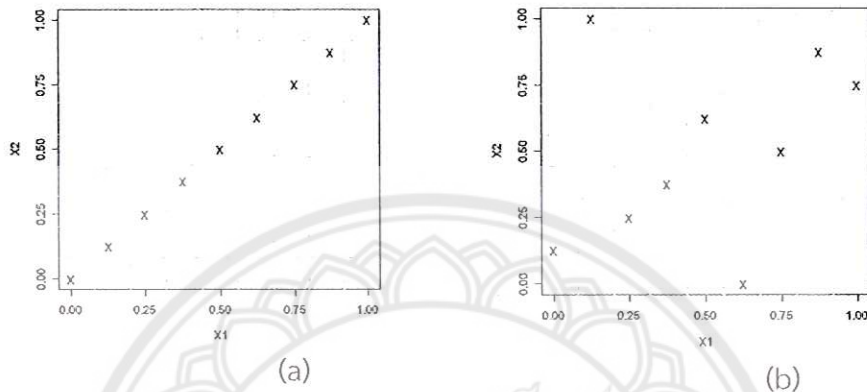
กล่าวโดยสรุปคือ งานวิจัยนี้จะมุ่งเน้นศึกษาหาปัจจัยที่มีผลต่อประสิทธิภาพของอัลกอริทึมการหาค่าที่เหมาะสมแต่ละประเภท จากนั้นจะมีการประยุกต์การวิเคราะห์ความไวเข้ามาใช้ เพื่อศึกษาหาเซตของพารามิเตอร์ที่เหมาะสมสำหรับอัลกอริทึมการหาค่าที่เหมาะสมแต่ละชนิด โดยผู้วิจัยคาดหวังว่าผลที่ได้จะสามารถนำมาสรุปเป็นตารางสำเร็จรูป ซึ่งผู้ใช้สามารถนำไปใช้ได้อย่างสะดวกรวดเร็ว จากนั้นผู้วิจัยจะนำเสนอแนวคิดในการปรับปรุงการทำงานของแต่ละอัลกอริทึมการหาค่าที่เหมาะสม เช่น การปรับโครงสร้างย่านใกล้เคียง การปรับกฎการแทนที่ การปรับกฎการหยุดสืบทอด และการปรับวิธีการคำนวณเกณฑ์การเลือกค่าที่เหมาะสม เป็นต้น เพื่อให้อัลกอริทึมแต่ละประเภท สามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ไม่ว่าจะเป็น ในแง่ของอัตราความเร็วในการลู่เข้าหาค่าที่เหมาะสม (Rate of convergence) รวมไปถึง การรายงานแผนการทดลองที่เหมาะสมที่ยังคุณสมบัติที่ดีเท่าเดิมทุกประการ แต่ใช้เวลาทดลองอย่างมีนัยสำคัญทางสถิติ หลังจากทำการปรับปรุงโครงสร้างการทำงานของอัลกอริทึมการหาค่าที่เหมาะสมแต่ละประเภทแล้ว ก็จะมีการเปรียบเทียบประสิทธิภาพของอัลกอริทึมประเภทต่างๆ เมื่อกำหนดมิติแผนการทดลองที่แตกต่างกัน เพื่อเป็นแนวทางในการเลือกใช้ อัลกอริทึมที่เหมาะสมสำหรับแต่ละปัญหา

การทบทวนวรรณกรรม/สารสนเทศ (Information) ที่เกี่ยวข้อง

การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiments: CSE) ได้ถูกนำมาใช้อย่างแพร่หลายเพื่อศึกษารูปแบบของระบบที่มีความซับซ้อนและไม่สามารถทำการทดลองทางกายภาพ (Physical experiment) ทั่วไปได้ Sacks et al. [31] รายงานว่าเวลาที่ใช้ในกระบวนการจำลองการทดลองด้วยคอมพิวเตอร์ มักจะใช้เวลานานมาก เช่น การจำลองคอมพิวเตอร์ในงานด้านวงจรรีเลย์อิเล็กทรอนิกส์อาจใช้เวลาเป็นวันในการหาค่าตัวแปรออกเพียง 1 ค่า ดังนั้นเพื่อเป็นการลดปัญหาที่เกิดขึ้น จึงได้มีการใช้ทฤษฎีทางสถิติเข้ามาใช้ในงานด้าน CSE ซึ่งประกอบด้วยการสร้างแผนการทดลองที่เหมาะสมบนคลาสของละตินไฮเปอร์คิวบ์ (Optimal Latin hypercube design: OLHD) และการพัฒนาตัวแบบทางสถิติเพื่อใช้ในการพยากรณ์ โครงการวิจัยนี้ผู้วิจัยจะเน้นศึกษาวิธีการสร้างแผนการทดลองที่เหมาะสมหรือ OLHD เพื่อใช้ใน CSE ซึ่ง Simpson et al. [34] ระบุว่าผลลัพธ์ที่ได้จาก CSE จะมีลักษณะเป็นแบบ Deterministic นั่นคือ การทำการทดลองที่ค่าตัวแปรเข้าที่คงที่ จะก่อให้เกิดค่าตัวแปรออกที่คงที่เสมอ ซึ่งจะเห็นได้ว่า การทำซ้ำ (Replicate) จะไม่มีความเกี่ยวข้องกับ CSE ดังนั้นการวางแผนการทดลองจึงมุ่งเน้นเพื่อกระจายจุดทดลองให้ทั่วถึงปริภูมิของการทดลอง (Space filling design) ให้มากที่สุด ซึ่งแผนการทดลองที่ใช้อย่างแพร่หลายใน CSE คือ แผนการทดลองแบบละตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) ซึ่งแผนการทดลองนี้ถูกนำเสนอโดย McKay et al. [23] ในปี ค.ศ.1979 คุณสมบัติพิเศษของแผนการทดลองแบบ LHD คือจุดทดลองทุกจุดจะไม่มีทับซ้อนกันเมื่อฉายภาพ (Projection) จากแต่ละมิติของตัวแปรเข้าใด ๆ แผนการทดลองแบบ LHD คือเมตริกซ์ X ซึ่งประกอบไปด้วย n แถว และ d คอลัมน์ เมื่อ n คือจำนวนรัน (จุดทดลอง) ในแผนการทดลอง และ d คือจำนวนตัวแปรเข้าในแผนการทดลอง การสร้างแผนการทดลองแบบ LHD สามารถทำได้ดังสมการที่ (9.1)

$$X_{ij} = \frac{\pi_{ij} - U_{ij}}{n} \quad (4)$$

เมื่อ π_{ij} เป็นตัวเลขที่เกิดจากการสลับตัวเลขแบบสุ่มที่ได้จากการแจกแจงแบบเอกรูป (Uniform distribution) จาก 1 ถึง n และ U_{ij} มีการแจกแจงแบบ $U[0,1]$ ซึ่งในทางปฏิบัติ การสร้าง LHD สามารถทำได้โดยการ สับเปลี่ยนแบบสุ่ม (Random permutation) ตัวเลข 1 ถึง n ในแต่ละคอลัมน์ จากนั้นนำตัวเลขที่ได้จากทั้งหมด d คอลัมน์มาผนวกเข้าด้วยกันเป็นเมตริกซ์ของแผนการทดลอง X ซึ่งวิธีการนี้จะทำให้การกระจายของแต่ละจุดทดลองจะเป็นไปโดยสุ่ม และไม่สามารถรับรองได้ว่าการกระจายของจุดจะทั่วถึงปริภูมิของการทดลองหรือไม่ ซึ่งสามารถพิจารณาได้จากรูปที่ 2



ภาพ 2 แสดงแผนการทดลอง 9×2 LHD แบบสุ่ม

จากรูปที่ 2(a) จะเห็นได้ว่าแผนการทดลองแบบ LHD ที่ได้จากการสุ่มตามวิธีการข้างต้นไม่มีคุณสมบัติที่ดี คือขาดคุณสมบัติเชิงตั้งฉาก ส่วนรูปที่ 2(b) แม้ว่าจะมีคุณสมบัติที่ดีขึ้นแต่จะเห็นได้ว่ามีบางพื้นที่ของปริภูมิของการทดลอง (Design space) อาจจะไม่ถูกสำรวจก็ได้ เพื่อทำการกระจายจุดทดลองให้ทั่วปริภูมิของแผนการทดลอง นักวิจัยได้ประยุกต์ใช้อัลกอริทึมการสืบค้น (Search algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าเหมาะสม (Optimality criteria) เพื่อสร้างแผนการทดลองที่เหมาะสมและมีคุณสมบัติความเต็มเติมปริภูมิและเชิงตั้งฉาก ดังที่ได้รายงานไว้ในงานวิจัยหลายชิ้น [9, 11, 12, 13, 16, 17, 18, 19, 24] ซึ่งรายงานไว้ว่า OLHD เป็นแผนการทดลองที่มีประสิทธิภาพดีกว่าแผนการทดลองแบบ Simple random sequence ทั่ว ๆ ไป แต่ทั้งนี้การค้นหาแผนการทดลอง LHD ที่เหมาะสม หรือ OLHD นั้นไม่ใช่เรื่องง่าย โดยเฉพาะอย่างยิ่งในกรณีที่มีมิติของแผนการทดลองมีขนาดใหญ่ขึ้น [5, 15, 25, 20, 38] ตัวอย่างบางส่วนของงานวิจัยที่ได้เสนอการสร้างแผนการทดลองแบบ OLHD โดยใช้อัลกอริทึมการสืบค้นควบคู่กับเกณฑ์เลือกค่าเหมาะสม เช่น การค้นหาแผนการทดลองที่เหมาะสม โดยใช้อัลกอริทึมการสืบค้นควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสมนั้นได้ถูกนำเสนอไว้หลายวิธี เช่น Morris and Mitchell [25] ได้พัฒนาอัลกอริทึม Simulated annealing (SA) ภายใต้เกณฑ์ ϕ_p จากนั้น Li and Wu [20] ได้เสนอการค้นหาแผนการทดลองที่เหมาะสมที่สุดโดยใช้อัลกอริทึม CP ภายใต้เกณฑ์ Integrated mean square error (IMSE) และ เกณฑ์เอ็นโทรปี (Entropy criteria) ต่อมา Jin et al. [15] ได้ดัดแปลงอัลกอริทึม วิวัฒนาการแบบสุ่ม (Enhanced stochastic evolutionary algorithm) เพื่อทำการค้นหาแผนการทดลองที่เหมาะสมโดยพิจารณาเกณฑ์ที่หลากหลายเช่น เกณฑ์ Maximin distance เกณฑ์ ϕ_p และเกณฑ์ Entropy ในปีถัดมา Liefvendahl and Stocki [21] ประยุกต์อัลกอริทึมแบบเจเนติก (Genetic algorithm: GA) เพื่อทำการค้นหาแผนการทดลองที่ดีที่สุด ภายใต้เกณฑ์ ϕ_p และ เกณฑ์ Maximin จากงานวิจัยของ Rungtattanabool and Na-udom [30] ได้ทำการเปรียบเทียบประสิทธิภาพของ SA และ GA ในแง่ของความเร็วในการลู่เข้าสู่ค่าที่เหมาะสม (Rate of convergence) และพบว่า SA สามารถทำงานได้ดีกว่า GA เกือบทุกกรณีของการศึกษา นอกจากนี้ Joseph and Hung [17] เสนอการสร้างแผนการทดลองแบบ LHD โดยพิจารณาเกณฑ์ Maximin

distance และเกณฑ์ Minimum pair-wise correlation ควบคุมกันโดยใช้อัลกอริทึมที่พัฒนาขึ้นมาใหม่และเปรียบเทียบผลที่ได้กับงานวิจัยที่ได้รายงานไว้ใน Morris and Mitchell [25] พบว่าประสิทธิภาพของแผนการทดลองในเชิงของคุณสมบัติที่พิจารณาพอ ๆ กัน งานของ Cioppa and Lucas [5] ได้เสนออัลกอริทึมสำหรับสร้างแผนการทดลอง OLHD ที่มีมิติใหญ่ขึ้นเมื่อเทียบกับงานวิจัยก่อนหน้า นอกจากนี้ยังพบว่าวิธีที่นำเสนอขึ้นมาก่อให้เกิดแผนการทดลองที่มีคุณสมบัติที่ดีมากในแง่ของความเต็มเต็มปริภูมิและคุณสมบัติเชิงตั้งฉาก จากนั้น Steinberg [35] เสนอการวางแผนการทดลองแบบ Orthogonal LHD โดยพยายามทำให้ค่าสหสัมพันธ์ระหว่างคอลัมน์ของแผนการทดลองมีค่าน้อยที่สุด ซึ่งเป็นการขยายงานที่นำเสนอไว้ใน Ye [37] เพื่อให้สามารถสร้างแผนการทดลองที่มีมิติใหญ่ขึ้นได้ ส่วน Prescott [29] เสนอการสร้างแผนการทดลองแบบ LHD ที่มีค่า pair-wise correlation เท่ากับ 0 และพิจารณาว่าคุณสมบัติดังกล่าวมีความสัมพันธ์กับคุณสมบัติความเต็มเต็มปริภูมิหรือไม่อย่างไรก็ตามข้อจำกัดของงานวิจัยชิ้นนี้คือวิธีการที่นำเสนอสามารถทำได้เฉพาะกรณีมิติของปัญหาขนาดเล็กเท่านั้น ในส่วนของการเสนออัลกอริทึมเพื่อสร้างแผนการทดลอง OLHD สามารถดูได้จาก Grosso et al. [12] เสนอวิธีสร้างแผนการทดลองแบบ LHD ที่เหมาะสมโดยใช้ Iterated Local Search (ILS) ซึ่งผลที่ได้พบว่าวิธีการดังกล่าวสามารถทำได้ดีเมื่อเปรียบเทียบกับงานที่เสนอไว้ใน Morris and Mitchell [25] และงานของ Jin et al. [15]

จากที่กล่าวมาข้างต้น จะเห็นได้ว่าการพัฒนาวิธีการสร้างแผนการทดลองแบบ OLHD มีความจำเป็นอย่างยิ่ง ซึ่งโครงการวิจัยนี้จะเน้นการผสมผสานข้อเด่นและข้อด้อยของการวางแผนการทดลองบนหลักการพีชคณิตและการใช้อัลกอริทึมการสืบค้นเข้าด้วยกัน และประยุกต์ใช้หลักการของ Iterated Local Search เข้ามาช่วยเพื่อให้ได้อัลกอริทึมที่สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น นอกจากนี้ การศึกษาสำรวจเบื้องต้นและการทบทวนงานวิจัยที่เกี่ยวข้อง พบว่าวิธีการค้นหาแผนการทดลองที่เหมาะสมดังกล่าวข้างต้น ได้ถูกนำมาใช้อย่างแพร่หลายในงานด้าน CSE แต่ยังมีข้อจำกัดอยู่มาก เช่น กระบวนการสืบค้นใช้เวลายาวนานมาก และบางกรณีการสืบค้นไม่สามารถค้นหาแผนการทดลองที่เหมาะสมที่เป็น Global optimum ได้ โดยเฉพาะอย่างยิ่งเมื่อมิติของปัญหา (n และ d) มีขนาดใหญ่ แต่ก็ได้มีการศึกษาหาทางแก้ไขและปรับปรุงให้อัลกอริทึมเหล่านี้สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น โดยการปรับปรุงเงื่อนไขต่าง ๆ ระหว่างการค้นหาแผนการทดลองที่เหมาะสม ดังนั้นงานวิจัยนี้จึงมุ่งเน้นที่จะพัฒนาอัลกอริทึมการสืบค้นที่มีประสิทธิภาพมากยิ่งขึ้นเพื่อสร้างแผนการทดลองที่เหมาะสมสำหรับ CSE บนคลาสของ LHD หรือ OLHD โดยเน้นการปรับกลไกภายในกระบวนการสืบค้น เช่น กฎการแทนที่ (Replacement rule) กฎการหยุดสืบค้น (Stopping rule) เป็นต้น ซึ่งผลที่ได้จากโครงการวิจัยนี้บางส่วนจะถูกนำไปประยุกต์ใช้จริงกับการวางแผนการทดลองด้านการแยกน้ำมันออกจากน้ำเสียซึ่งเป็นโครงการวิจัยภายใต้ความร่วมมือระหว่างข้าพเจ้าและนักวิจัยจากภาควิชาวิศวกรรมเคมี มหาวิทยาลัยพระจอมเกล้าพระนครเหนือ

วัตถุประสงค์ของการวิจัย

1. เพื่อนำเสนออัลกอริทึมการสืบค้นที่มีประสิทธิภาพในการสร้างแผนการทดลองที่เหมาะสมสำหรับ CSE
2. เพื่อเสนอแนวทางในการปรับปรุงการทำงานของอัลกอริทึมการสืบค้นให้มีประสิทธิภาพและยืดหยุ่นต่อการนำไปใช้งาน
3. เพื่อนำแผนการทดลองที่เหมาะสมที่ได้จากอัลกอริทึมการสืบค้นไปใช้กับการจำลองทางด้านวิศวกรรมเคมี

ประโยชน์ที่คาดว่าจะได้รับ

1. งานวิจัยนี้จะก่อให้เกิดวิธีการสร้างแผนการทดลองโดยใช้อัลกอริทึมการสืบค้นที่มีประสิทธิภาพมากยิ่งขึ้นและสามารถใช้งานได้ง่ายขึ้น และจะได้ข้อสรุปใหม่ ๆ เกี่ยวกับการสร้างแผนการทดลองที่มีมิติของปัญหาใหญ่ขึ้น
2. ทำให้ทราบถึงแนวทางการปรับค่าพารามิเตอร์ที่เกี่ยวข้องกับอัลกอริทึมการสืบค้นเพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น
3. พัฒนาศักยภาพนักวิจัยในโครงการและผลิตผลงานเพื่อตีพิมพ์ในวารสารระดับนานาชาติ เช่น Journal of Statistical Planning and Inference หรือ Journal of global optimization ได้
4. ทำให้เกิดความร่วมมือในการทำวิจัย ระหว่างนักวิจัยสังกัดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ โดย รศ.ดร. จันทพร ผลากรกุล และนักวิจัยจากมหาวิทยาลัยนเรศวรเพื่อการพัฒนางานวิจัยอย่างต่อเนื่องในอนาคต

ขอบเขตการวิจัย

โครงการวิจัยนี้เน้นการศึกษาข้อจำกัดของแผนการทดลองที่ใช้อย่างแพร่หลายในงานด้านการจำลองการทดลองด้วยคอมพิวเตอร์ หรือ CSE จากนั้นจะทำการปรับปรุงข้อจำกัดเหล่านั้นเพื่อให้กระบวนการวางแผนทดลองสำหรับ CSE มีประสิทธิภาพมากยิ่งขึ้น นอกจากนี้ยังมีวัตถุประสงค์ในการขยายขอบเขตของการศึกษาต่าง ๆ ที่รายงานไว้ในเอกสารที่เกี่ยวข้อง เพื่อหาข้อสรุปใหม่ ๆ รวมไปถึงการนำเสนอวิธีการใหม่ ๆ เพื่อให้สามารถวางแผนทดลองที่ดียิ่งขึ้น หรืออย่างน้อยต้องมีประสิทธิภาพเท่าเดิม แต่สามารถใช้งานได้ง่ายกว่าของเดิมที่มีอยู่แล้ว โครงการวิจัยนี้จะเน้นการพัฒนาการสร้างแผนการทดลองที่เหมาะสมบนคลาสของ LHD โดยการผสมผสานแนวคิดของหลักการพีชคณิตและอัลกอริทึมการสืบค้นเข้าด้วยกัน และปรับกลไกการทำงานของอัลกอริทึมการสืบค้นให้มีประสิทธิภาพมากยิ่งขึ้นโดยพิจารณาจากผลลัพธ์ที่ได้จากการจำลองสถานการณ์ภายใต้มิติที่แตกต่างกัน ดังรายละเอียดต่อไปนี้

1) ประเภทของแผนการทดลอง

1.1) แผนการทดลองทางกายภาพแบบต่างๆ เช่น แฟรคชันแนลแฟคทอเรียล (Fractional factorial design: FFD) แผนการทดลองแบบ Central composite design (CCD) แบบต่างๆ เป็นต้น

1.2) แผนการทดลองแบบ LHD ประเภทต่าง ๆ เช่น Orthogonal LHD, Symmetric LHD

2) ประเภทของอัลกอริทึมการสืบค้น

2.1) Columnwise-pairwise algorithm (CP)

2.2) Simulated annealing algorithm (SA)

2.3) Iterated Local Search (ILS)

3) เกณฑ์การเลือกค่าที่เหมาะสมที่สุด

3.1) Maximin distance criterion

3.2) ϕ_p criterion

3.3) Minimum pair-wise correlation

วิธีดำเนินการวิจัย

- 1) ศึกษาและค้นคว้าทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง
- 2) สร้างแผนการทดลองที่เหมาะสมโดยใช้วิธีการพีชคณิตและวิธีการประมาณด้วยอัลกอริทึมการหาค่าเหมาะสม
- 3) ปรับวิธีการกำหนดค่าดั้งเดิม กฎการแทนที่ และการหาชุดของพารามิเตอร์ที่เหมาะสมของแต่ละอัลกอริทึมการสืบค้น
- 4) พัฒนาโปรแกรม R เพื่อจำลองกรณีปัญหาต่าง ๆ
- 5) รันโปรแกรมเพื่อเก็บรวบรวมข้อมูล และทำการวิเคราะห์ข้อมูลเพื่อหาข้อสรุป
- 6) สรุปผลและจัดทำบทความเพื่อตีพิมพ์เผยแพร่



บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้ผู้วิจัยจะกล่าวถึงวิธีการพัฒนาอัลกอริทึมการสืบค้นแบบสโทแคสติกเชิงวิวัฒนาการ โดยใช้ภาษา R ซึ่งประกอบด้วย การแลกเปลี่ยนสมาชิก การปรับปรุงอัลกอริทึมการสืบค้นการประเมินค่า Φ_p แบบใหม่ และการวิเคราะห์กฎการหยุดสืบค้นที่เหมาะสม

การแลกเปลี่ยนสมาชิก (Element exchange operation)

การสร้างแผนการทดลองใหม่ X_{try} ที่ใกล้เคียงกับแผนการทดลองปัจจุบัน X โดยใช้แนวคิดจากปฏิบัติการเชิงคอลัมน์ (Column - wise operations) นำเสนอโดย Li and Wu (1997) วิธีการ คือ สุ่มแลกเปลี่ยนสมาชิกในคอลัมน์ที่ k และแถวที่ a และ b ซึ่งการแลกเปลี่ยนสมาชิกนี้จะทำให้แผนการทดลอง X_{try} ยังคงคุณสมบัติของ LHD และมีข้อดีคือทำให้สามารถใช้วิธีประเมินค่า Φ_p แบบใหม่สำหรับแผนการทดลอง X_{try} ได้ อย่างมีประสิทธิภาพ (Jin, et al., 2005) ตัวอย่างการแลกเปลี่ยนสมาชิกแสดงในภาพ 9

0	0.125	0	0.125
0.25	0.25	0.25	0.25
0.375	0.375	0.375	0.5
0.5	0.625	0.5	0.625
0.625	0	0.625	0
0.75	0.5	0.75	0.375
0.875	0.875	0.875	0.875
1	0.75	1	0.75
0.125	1	0.125	1

ภาพ 3 การแลกเปลี่ยนสมาชิกของแผนการทดลองแบบ LHD ขนาด 9×2

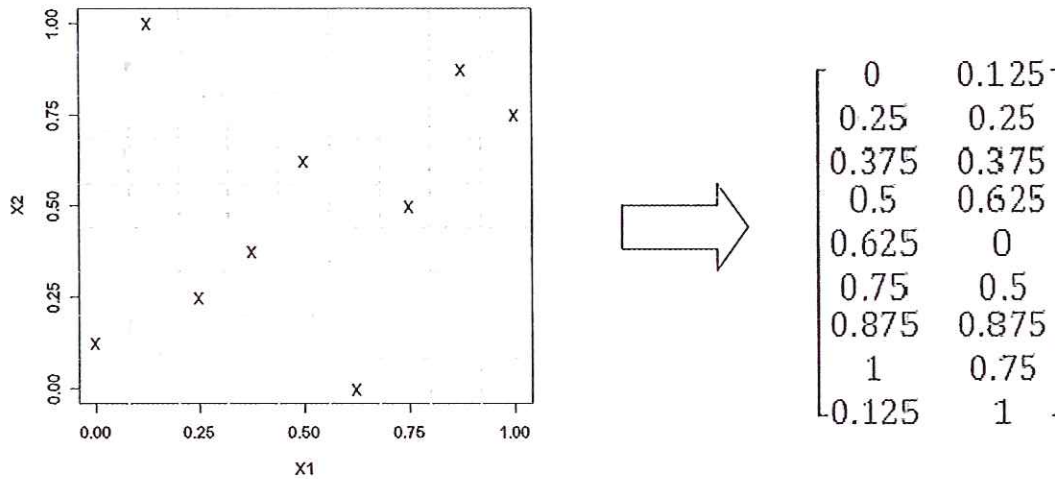
ภาพ 9 แสดงการแลกเปลี่ยนสมาชิกโดยมี $k = 2, a = 3, b = 6$ จากแผนการทดลองเริ่มต้น (ด้านซ้ายมือ) สมาชิกในคอลัมน์ที่ 2 แถวที่ 3 คือ 0.375 จะถูกสลับที่กับสมาชิกในคอลัมน์ที่ 2 แถวที่ 6 คือ 0.5 เปลี่ยนไปเป็นแผนการทดลองใหม่ (ด้านขวามือ)

การเพิ่มประสิทธิภาพอัลกอริทึมการสืบค้น

การนำทฤษฎีที่มีอยู่มาปรับใช้เพื่อเพิ่มประสิทธิภาพของอัลกอริทึมการสืบค้น เช่น การปรับปรุงวิธีการค้นหาแบบเชิงกลุ่มเพื่อให้การค้นหาแผนการทดลองสำหรับ CSE มีประสิทธิภาพมากขึ้น

1. Modified simulated annealing (Thamma, et al., 2008)

การพัฒนาอัลกอริทึม Modified simulated annealing (MSA) ขึ้นโดยใช้ตัวแปลภาษา R เพื่อใช้ในการค้นหาแผนการทดลองสำหรับ CSE ดังภาพ 11 สามารถอธิบายขั้นตอนการทำงานและวิธีการตั้งค่าพารามิเตอร์เริ่มต้นดังนี้



ภาพ 40 การสร้างแผนการทดลองแบบ LHD เริ่มต้นขนาด 9×2

ขั้นตอนที่ 1 สร้างแผนการทดลองเริ่มต้น X_0 โดยการสุ่มเรียงสับเปลี่ยนค่าที่เป็นไปได้ของตัวแปรในแผนการทดลองตามหลักการของ LHD ดังภาพ 10 ให้ค่ากับแผนการทดลองปัจจุบัน X และ ทำการบันทึกแผนการทดลอง X ให้เป็นแผนการทดลองที่ดีที่สุด (Best so far) $X_{best} = X$ เพื่อใช้สำหรับเปรียบเทียบกับแผนการทดลองใหม่ X_{try} และตั้งค่าพารามิเตอร์เริ่มต้นได้แก่ อุณหภูมิ t ที่เป็นเสมือนพารามิเตอร์ควบคุมตามสมการค่าเฉลี่ยของระยะทางระหว่างจุดทุกคู่ของแผนการทดลอง X_0 ดังนี้ จำนวนรอบการสร้างแผนการทดลองใหม่ i_{max} ซึ่งค่านี้จะต้องมากเพียงพอที่จะทำให้อัลกอริทึม MSA หลุดจากการหาค่าเฉลี่ยเฉพาะที่ได้ โดย i_{max} ถูกตั้งค่าตามจำนวนของตัวแปรเข้า ในแผนการทดลองคือ 1000 สำหรับจำนวนตัวแปรที่มากกว่าหรือเท่ากับ 6 และ 500 สำหรับจำนวนตัวแปรตั้งแต่ 7 ตัวขึ้นไป อัตราส่วนสำหรับลดอุณหภูมิ $c_t \leq 0.95$ เพื่อให้การลดอุณหภูมิ t ในกระบวนการค้นหาเป็นไปอย่างช้าๆ ค่าผลต่างของการยอมรับแผนการทดลองใหม่ $tol = 0.0001$ ซึ่งจากการศึกษาพบว่าถ้า tol น้อยเกินไปจะทำให้อัลกอริทึม MSA ใช้เวลาในการทำงานนานขึ้นแต่ไม่ทำให้ผลเฉลี่ยที่ได้ดีขึ้น

ขั้นตอนที่ 2 ตั้งค่าพารามิเตอร์สำหรับ ตรวจสอบการหยุดค้นหา $flag = FALSE$ และ $i = 1$

ขั้นตอนที่ 3 สร้างแผนการทดลองใหม่ X_{try} โดยการสุ่มแลกเปลี่ยนสมาชิกในคอลัมน์ k ของแผนการทดลอง $X_{(a,k)} \leftrightarrow X_{(b,k)}$ ดังภาพ 8

ขั้นตอนที่ 4 ประเมินค่าฟังก์ชันวัตถุประสงค์ Φ_p ของแผนการทดลอง X และ X_{try} เพื่อตัดสินใจว่าจะยอมรับ (Accept) แผนการทดลอง X_{try} แทนที่แผนการทดลอง X หรือไม่ตามสมการ (10)

$$\Phi_p(X) - \Phi_p(X_{try}) \geq tol$$

โดย $tol = 0.0001$ หรือยอมรับแผนการทดลอง X_{try} ตามความน่าจะเป็นในสมการ

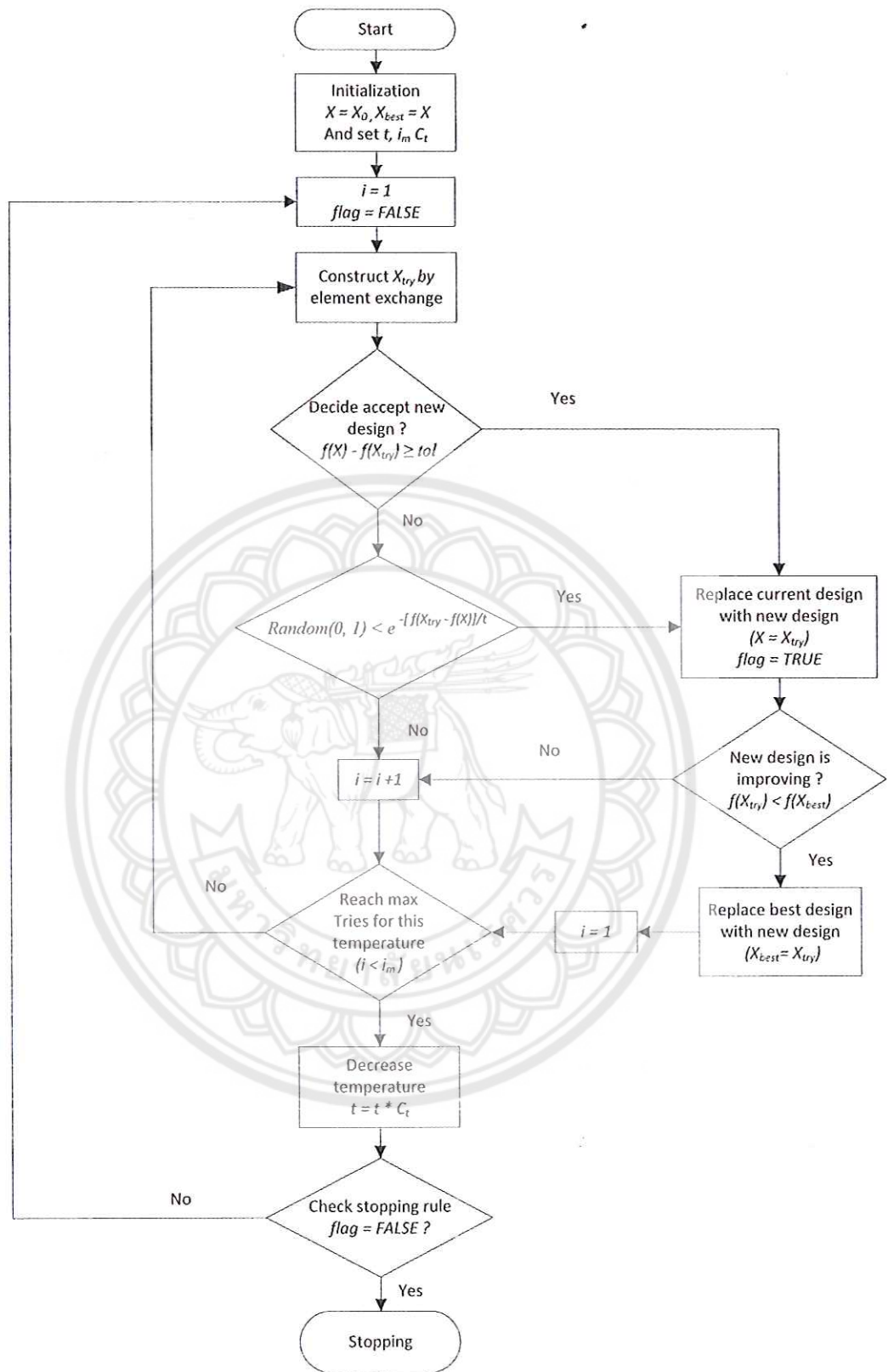
$$e^{-\left[\frac{\Phi_p(X_{try}) - \Phi_p(X)}{t}\right]}$$

ถ้าแผนการทดลอง X_{try} ถูกยอมรับตามเงื่อนไขใดเงื่อนไขหนึ่งต้องเปลี่ยนค่า $flag = TRUE$ แล้วเปรียบเทียบแผนการทดลอง X_{try} กับแผนการทดลองที่ดีที่สุด X_{best} ถ้า $\Phi_p(X_{try}) < \Phi_p(X_{best})$ แล้วจะทำการบันทึกแผนการทดลอง X_{try} ให้เป็นแผนการทดลองที่ดีที่สุด $X_{best} = X_{try}$

ขั้นตอนที่ 5 ขั้นตอนการทำงานจะวนซ้ำกลับไปยังขั้นตอนที่ 3 อีกครั้งจนครบจำนวนรอบเท่ากับ i_m แล้วจึงทำการลดอนุกรมคูณด้วยสมการ $t = t * c_t$

ขั้นตอนที่ 6 ขั้นตอนการทำงานจะวนซ้ำกลับไปขั้นตอนที่ 2 จนกว่าจะเป็นไปตามเงื่อนไขการหยุดทำงานของอัลกอริทึม $flag = FALSE$ เป็นจริง

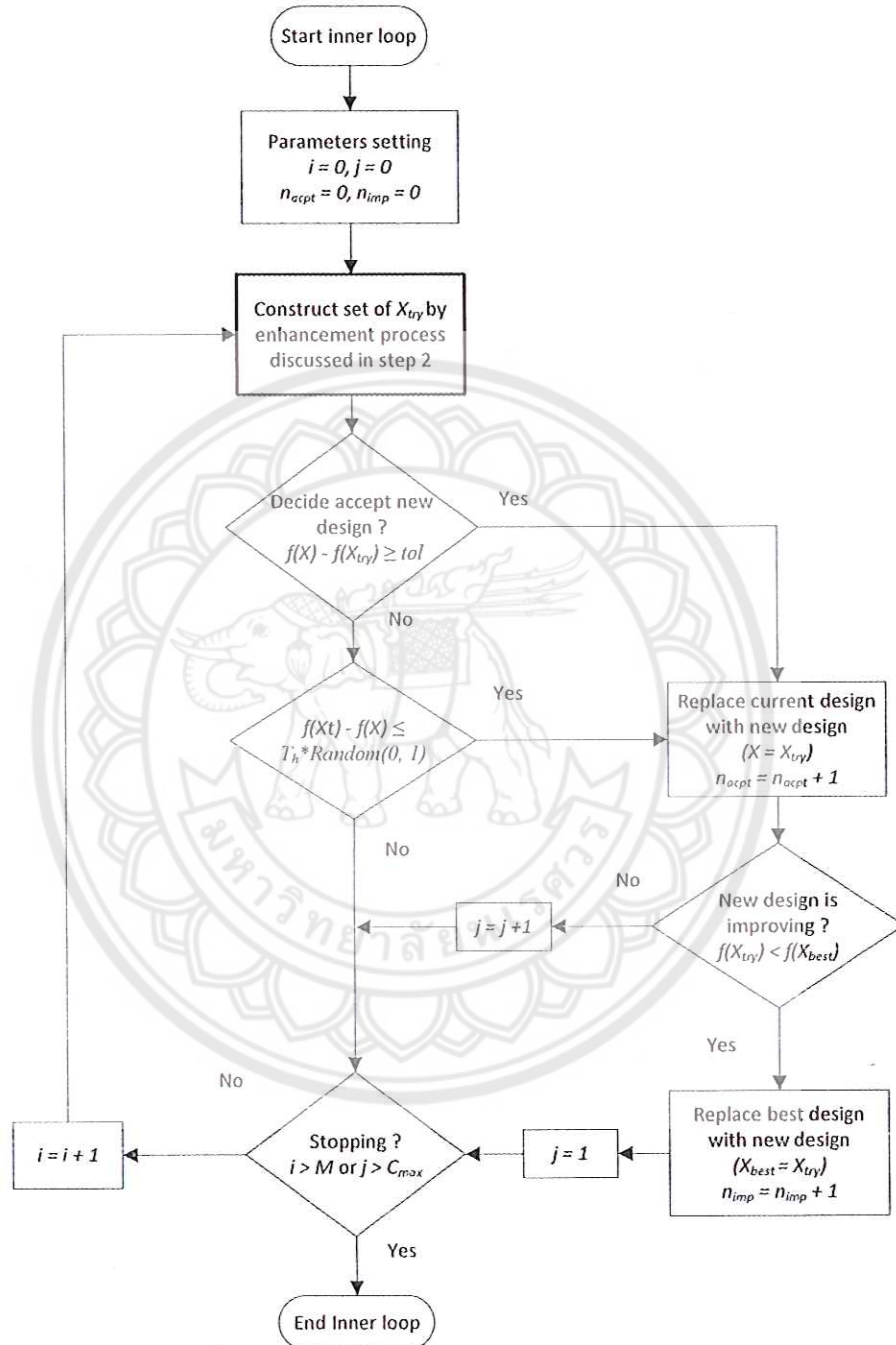




ภาพ 5 Modified simulated annealing algorithm flowchart

2. Enhancement of enhanced stochastic evolutionary

อัลกอริทึม Enhancement of enhanced stochastic evolutionary พัฒนาขึ้นโดยใช้ตัวแปลภาษา R เช่นเดียวกับ MSA



ภาพ 7 Enhancement of enhanced stochastic evolutionary flowchart (ลูปใน)

ผู้วิจัยได้พัฒนา MESE จากการนำเอาข้อดีของอัลกอริทึม MSA ที่นำเสนอโดย Thamma, et al. (2008) คือ ส่วนของวิธีการค้นหาแบบเฉพาะที่ ร่วมกับข้อดีของอัลกอริทึม Enhanced stochastic evolutionary (ESE) ที่นำเสนอโดย Jin, et al. (2005) ซึ่งได้อธิบายไปในบทที่ 2 ในส่วนของกระบวนการค้นหาเชิงกลุ่ม ที่สามารถปรับตัวเอง และเลือกกระบวนการค้นหาได้ และเพิ่มเติมในส่วนของการวิเคราะห์เงื่อนไขการหยุดทำงานที่

เหมาะสมเข้าไป เพื่อให้การค้นหาแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์มีประสิทธิภาพมากขึ้น โดยโครงสร้างของอัลกอริทึม MESE ประกอบด้วยการทำงาน 2 ลูปซ้อนกัน เช่นเดียวกับ ESE การทำงานลูปในเป็นกระบวนการค้นหาเพื่อสร้างแผนการทดลองใหม่ X_{try} ดังภาพ 12 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

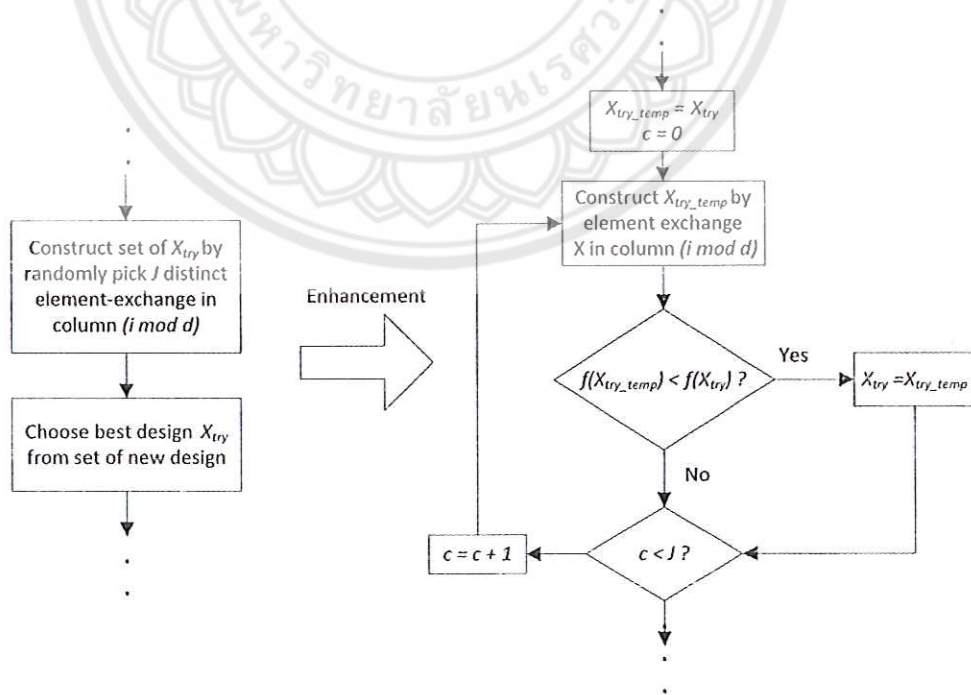
กลไกของลูปใน (Inner loop)

การทำงานของลูปใน ได้นำเอาข้อดีของอัลกอริทึม MSA มาประยุกต์ใช้ในการสร้างแผนการทดลองใหม่ X_{try} การตัดสินใจยอมรับแผนการทดลอง X_{try} รวมถึงเงื่อนไขการหยุดการทำงานลูปใน โดยลูปในจะมีการทำงานวนซ้ำทั้งหมด M รอบ หรือหยุดเมื่อแผนการทดลอง X_{try} ไม่ดีขึ้นจาก X_{best} ติดต่อกัน C_{max} รอบ

ขั้นตอนที่ 1 ก่อนเข้าสู่การทำงานลูปในต้องมีการทำงานในส่วนของลูปนอกซึ่งครอบคลุมอยู่คือ การสร้างแผนการทดลองเริ่มต้น X_0 และเข้าสู่การทำงานของลูปคือตั้งค่าเริ่มต้นของพารามิเตอร์ $i = 0, n_{accept} = 0, n_{imp} = 0$ และ C_{max}

ขั้นตอนที่ 2 สร้างแผนการทดลอง ใหม่ X_{try} โดยใช้วิธีการของอัลกอริทึม MSA ซึ่งมีผลทำให้ใช้เวลาในการสร้างแผนการทดลองน้อยลง และแผนการทดลองที่ได้หลังจบการทำงานลูปในเปลี่ยนแปลงไปไม่มาก (แสดงผลการเปรียบเทียบในส่วนของผลการวิจัย) โดยมีขั้นตอนการพัฒนาและการพิสูจน์การลดระยะเวลาการคำนวณได้ดังนี้

การพัฒนาวิธีการค้นหาโดยการปรับวิธีการสร้างกลุ่มของแผนการทดลองใหม่ X_{try} จากเดิมที่สุ่มสร้างแผนการทดลองที่ไม่ซ้ำกันจำนวน j แบบ โดยเปลี่ยนใช้วิธีการสร้างแผนการทดลองตามแบบของอัลกอริทึม MSA ที่แต่ละรอบการแลกเปลี่ยนสมาชิกเป็นอิสระต่อกัน ดังภาพ 13 ทำให้สามารถช่วยลดระยะเวลาในการสร้างแผนการทดลองซึ่งอธิบายได้ ดังนี้



ภาพ 8 การพัฒนาวิธีสร้างแผนการทดลองใหม่ในขั้นตอนที่ 2 ของลูปใน



การสุ่มข้อมูลจำนวน j แบบที่ไม่ซ้ำกันนั้นมีขั้นตอนการทำงานคล้ายกับอัลกอริทึมการเรียงข้อมูล (Sorting) ตัวอย่างเช่น การเรียงลำดับแบบแทรก (Insertion sort) และการเรียงลำดับแบบเลือก (Selection sort) กล่าวคือ การเรียงข้อมูลซึ่งต้องมีการนำข้อมูลตั้งแต่ลำดับที่ 1 จนถึงลำดับที่ n เปรียบเทียบกับข้อมูลในอาร์เรย์ที่มีข้อมูล n ข้อมูลซึ่งมีความซับซ้อนในการคำนวณ (Computation complexity) เท่ากับ $O(n^2)$ ในทำนองเดียวกัน ทุกครั้งของการสุ่ม ต้องมีการตรวจสอบว่ารอบการสุ่มนั้นได้ข้อมูลที่มีอยู่แล้วในอาร์เรย์ (Array) หรือไม่ดังนี้

การสุ่มครั้งที่ 1 ตรวจสอบกับข้อมูลใน อาร์เรย์ 0 ครั้ง
การสุ่มครั้งที่ 2 ตรวจสอบกับข้อมูลใน อาร์เรย์ 1 ครั้ง

๑ ๑A
๗๖
๑.๙๕๓
๑1615
๒๕๕๖

การสุ่มครั้งที่ $j - 1$ ตรวจสอบกับข้อมูลใน อาร์เรย์ $j - 2$ ครั้ง
การสุ่มครั้งที่ j ตรวจสอบกับข้อมูลใน อาร์เรย์ $j - 1$ ครั้ง

จากวิธีการสุ่ม ที่ได้กล่าวมาทำให้มีความซับซ้อน ในการคำนวณเท่ากับ $O(n * (n - 1)) = O(n^2)$ ซึ่งใช้เวลานานเมื่อแผนการทดลองหรือจำนวนการรันเพิ่มมากขึ้น แต่สำหรับวิธีการสร้างแผนการทดลองของ MSA ใช้วิธีการวนรอบเพื่อสร้างแผนการทดลองโดยการแลกเปลี่ยนสมาชิกจำนวน j รอบ แต่ละรอบมีความอิสระต่อกัน ตามที่ได้อธิบายได้ในส่วนของขั้นตอนการทำงานของอัลกอริทึม MSA ซึ่ง มีความซับซ้อน ในการคำนวณเท่ากับ $O(n)$ เท่านั้น

ขั้นตอนที่ 3 การตัดสินใจยอมรับแผนการทดลอง X_{try}

การพัฒนาวิธีการยอมรับแผนการทดลอง X_{try} จากเดิมที่ยอมรับแผนการทดลอง X_{try} ถ้า $\Phi_p(X_{try}) < \Phi_p(X)$ หรือ ยอมรับด้วยความน่าจะเป็นตามสมการ (13) โดยการปรับไปใช้วิธีการตัดสินใจยอมรับแผนการทดลองของ MSA ซึ่งทำให้สมการในการยอมรับแผนการทดลอง X_{try} มีความเคร่งครัดน้อยลง ดังนี้

$$\Phi_p(X) - \Phi_p(X_{try}) \geq tol$$

หรือ ยอมรับด้วยความน่าจะเป็นตามสมการ

$$\Phi_p(X_t) - \Phi_p(X) \leq T_h * random(0,1)$$

เมื่อ $random(0,1)$ ฟังก์ชันสำหรับสุ่มตัวเลขจำนวนจริงที่มีการกระจายแบบเอกรูป (Uniform) ระหว่าง

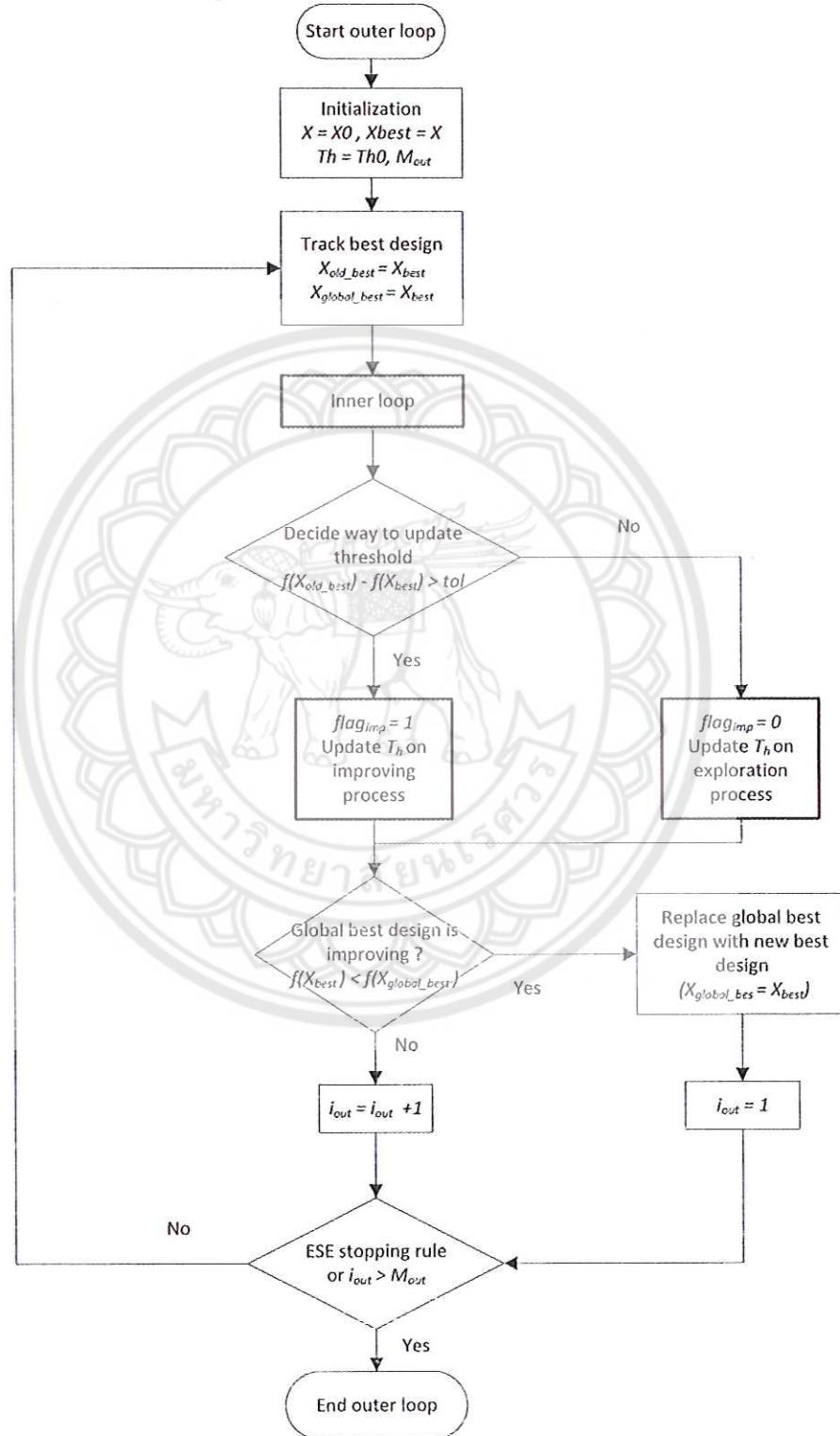
$(0, 1)$ และเพิ่มค่า $n_{acpt} = n_{acpt} + 1$ เมื่อแผนการทดลอง X_{try} ถูกยอมรับ

ขั้นตอนที่ 4 เปรียบเทียบแผนการทดลอง X_{try} กับแผนการทดลองที่ดีที่สุด X_{best} และถ้า X_{try} ดีกว่า X_{best} ก็จะทำให้การบันทึกค่า $X_{best} = X_{try}$ และ $n_{imp} = n_{imp} + 1$

ขั้นตอนที่ 5 การตรวจสอบเงื่อนไขการหยุดทำงาน

การพัฒนาวิธีการยอมรับแผนการทดลองในขั้นตอนที่ 3 นั้น สามารถช่วยลดจำนวนรอบการทำงานลูบใน โดยต้องทำไปควบคู่กับการเพิ่มเงื่อนไขการหยุดเข้าไปในกฎการหยุดสืบค้นของลูบในขั้นตอนที่ 5 ซึ่งถ้าตรงตาม

เงื่อนไขใดเงื่อนไขหนึ่งให้หยุดการทำงาน คือ $i > M$ ซึ่งในการวิจัยครั้งนี้ให้ค่า $M = 2\binom{n}{2}d/j$ แต่มีค่าไม่เกิน 100 หรือ $j > C_{max}$ ในการวิจัยครั้งนี้ให้ค่า $C_{max} = 10d + 10$ ซึ่งการใช้ตัวกระทำทางตรรกะ (Logical operator) หรือ (or) มาเชื่อมระหว่างเงื่อนไขการหยุดของทั้งสองเงื่อนไขที่กล่าวไป ทำให้สามารถแน่ใจได้ว่ารอบการทำงานของกลุ่มในจะไม่เพิ่มขึ้นจากอัลกอริทึม ESE เดิมอย่างแน่นอน



ภาพ 9 Enhancement of enhanced stochastic evolutionary flowchart (ลูปนอก)

กลไกของลูปนอก (Outer loop)

การทำงานของลูปนอกเป็นการนำเอาข้อดีของการค้นหาแบบเชิงกลุ่ม (Global search) ของอัลกอริทึม ESE มาใช้ ลูปนอกทำหน้าที่ควบคุมกระบวนการค้นหาโดยการปรับค่าพารามิเตอร์ควบคุม T_h ซึ่งในตอนเริ่มต้นการทำงานของอัลกอริทึมค่า T_h จะถูกตั้งให้ไว้ต่ำ (จะกล่าวถึงในส่วนของขั้นตอนการทำงาน) หลังจากนั้นจะถูกปรับค่าไปตามกระบวนการค้นหาซึ่งแบ่งได้เป็น 2 กรณี คือ ถ้าแผนการทดลองที่ได้หลังจากจบการทำงานของลูปในในรอบปัจจุบันดีขึ้นเมื่อเทียบกับรอบก่อนหน้าจะใช้วิธีการค้นหาแบบปรับปรุง (Improving process) เพื่อปรับแผนการทดลองให้มีคุณสมบัติดียิ่งขึ้น และถ้าแผนการทดลองที่ได้หลังจากจบการทำงานของลูปในในรอบปัจจุบันไม่ดีขึ้นเมื่อเทียบกับรอบก่อนหน้าจะใช้การค้นหาแบบสำรวจ (Exploration process) ซึ่งตรงนี้เป็นข้อดีของ ESE ที่นำมาควบคุมการทำงานในลูปนอก การทำงานในลูปนอกมีขั้นตอนดังนี้

ขั้นตอนที่ 1 สร้างแผนการทดลองเริ่มต้น X_0 ให้ค่ากับแผนการทดลองปัจจุบัน $X = X_0$ บันทึกแผนการทดลองที่ดีที่สุด $X_{best} = X$ และ ตั้งค่าของตัวแปรควบคุมเริ่มต้น T_{h0} ในตอนแรกตั้งให้น้อยๆ ตามสมการ (14)

$$T_h = T_{h0} = 0.005 \times \Phi_p(X_0) \quad (14)$$

ขั้นตอนที่ 2 บันทึกแผนการทดลองที่ดีที่สุดก่อนเข้าสู่การทำงานลูปใน $X_{old_best} = X_{best}$ เพื่อใช้เปรียบเทียบกับแผนการทดลองที่ดีที่สุดหลังจบการทำงานลูปใน เพื่อใช้ตรวจสอบเงื่อนไขการเลือกกระบวนการค้นหา และบันทึกแผนการทดลองที่ดีที่สุดเชิงกลุ่ม $X_{global_best} = X_{best}$

ขั้นตอนที่ 3 เข้าสู่การทำงานในส่วนของลูปในเมื่อจบการทำงานจะคืนกลับ (Return) ค่า X_{best} เพื่อใช้ในขั้นตอนต่อไป

ขั้นตอนที่ 4 การเลือกกระบวนการค้นหาโดยการตั้งค่า $flag_{imp}$ ตามเงื่อนไขของสมการ

$$\Phi_p(X_{old_best}) - \Phi_p(X_{best}) \geq tol$$

ขั้นตอนที่ 5 ตรวจสอบว่าแผนการทดลอง X_{best} มีคุณสมบัติที่ดีกว่า X_{global_best} หรือไม่ตามสมการ $\Phi_p(X_{best}) < \Phi_p(X_{global_best})$ ถ้าดีกว่าก็ทำการแทนที่แล้ว เพิ่มค่า

$$i_{out} = i_{out} + 1$$

กระบวนการปรับค่า T_h ทั้งหมดเหมือนกับ ESE (ภาพ 7 และภาพ 8) โดยสามารถอธิบายได้ดังนี้ การค้นหาแบบปรับปรุง (Improving process) ($flag_{imp} = 1$) การปรับค่า T_h มีพื้นฐานบนอัตราส่วน n_{acpt}/M และอัตราส่วน n_{imp}/M ตามเงื่อนไขดังนี้

1. T_h จะลดลงตามสมการ $T_h = \alpha_1 * T_h$ ถ้าอัตราส่วน $n_{acpt}/M > 10\%$ และอัตราส่วน $n_{imp}/M < n_{acpt}/M$

2. T_h จะไม่เปลี่ยนแปลงถ้าอัตราส่วน $n_{acpt}/M > 10\%$ และ อัตราส่วน $n_{imp}/M = n_{acpt}/M$

3. นอกเหนือจากสองเงื่อนไขแรกจะทำการเพิ่ม T_h ตามสมการ $T_h = T_h/\alpha_1$ โดยงานวิจัยนี้กำหนดให้ $\alpha_1 = 0.8$ และ $0 < \alpha_1 < 1$ (Jin, et al., 2005)

การค้นหาแบบสำรวจ (Exploration process) ($flag_{imp} = 0$) การปรับค่า T_h จะมีพื้นฐานอยู่บนช่วงของอัตราส่วน n_{acpt}/M ถ้า ตามเงื่อนไขดังนี้

1. T_h เพิ่มขึ้นตามสมการ $T_h = T_h/\alpha_3$ ถ้า $n_{acpt}/M < 10\%$
จนกระทั่ง $n_{acpt}/M > 80\%$

2. T_h ลดลงตามสมการ $T_h = T_h * \alpha_2$ ถ้า $n_{acpt}/M > 80\%$
จนกระทั่ง $n_{acpt}/M < 10\%$ โดยกำหนดให้ $\alpha_2 = 0.9, \alpha_3 = 0.7$ และ $0 < \alpha_2 < \alpha_3 < 1$ (Jin, et al., 2005)

ขั้นตอนที่ 6 ตรวจสอบเงื่อนไขการหยุดทำงานของ MESE

ขั้นตอนนี้ผู้วิจัยได้ทำการพัฒนาโดยการเพิ่มเงื่อนไขการหยุดเข้าไปอีก 1 เงื่อนไขในกฎการหยุดสืบค้นของ ESE เดิมทำให้อัลกอริทึมไม่ต้องสร้างแผนการทดลองให้ครบตามจำนวนรอบการทำงานที่ตั้งไว้ถ้าค่า Φ_p จากการทำงานของ MESE ไม่ดีขึ้นเป็นจำนวนหลายๆ รอบติดกัน (ภาพ 14)

การเปลี่ยนวิธีการประเมินคุณสมบัติแผนการทดลอง (Re - calculate Φ_p)

การแลกเปลี่ยนสมาชิกของแผนการทดลอง X เพื่อสร้างแผนการทดลอง X_{try} ด้วยวิธีที่กล่าวไปในตอนต้น สามารถใช้วิธีการประเมินค่า Φ_p แบบใหม่ซึ่งมีผลทำให้สามารถช่วยลดระยะเวลาในการคำนวณ จากเดิมที่การประเมินค่า Φ_p ต้องมีกระบวนการดังนี้

Φ_p เกณฑ์การเลือกค่าเหมาะสมที่นำเสนอโดย Morris and Mitchell (1995) ซึ่งปรับปรุงมาจาก Maximin distance สำหรับแผนการทดลอง X ระยะทางระหว่างจุดทดลองบนปริภูมิการทดลองแบบยูคลิด (Euclidean distance) สามารถหาโดยใช้สมการ (16)

$$d(X_j, X_k) = \left[\sum_{i=1}^d (X_{ji} - X_{ki})^2 \right]^{\frac{1}{2}} \quad (16)$$

เมื่อคำนวณหาระยะทางระหว่างจุดทุกคู่ในแผนการทดลองซึ่งเขียนอยู่ในรูปของเมทริกซ์จัตุรัส

$D = [d_{ik}]_{n \times n}$ ให้ทำการเรียงระยะทางระหว่างจุดทั้งหมดจากน้อยไปมาก (d_1, d_2, \dots, d_m) และสร้างดัชนี (Index) ($j_1, j_2, \dots, j_m, j_j$) คือ จำนวนของคู่จุดที่แบ่งตามระยะทาง d_j เกณฑ์การเลือกค่าเหมาะสม Φ_p สามารถคำนวณได้จากสมการ (17)

วิธีการคำนวณให้ง่ายขึ้นโดยไม่นำค่า $\Phi_p = \left[\sum_{j=1}^m j_j d_j^{-p} \right]^{\frac{1}{p}}$ าร (18) ซึ่งโดยทั่วไปแล้วแผนการทดลองแบบ LHD ที่เหมาะสมคือแผนการทดลองที่เหคา Ψ_p น้อยที่สุด

$$\Phi_p = \left[\sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{(d_{ij})^p} \right]^{\frac{1}{p}}$$

ซึ่งจะใช้วิธีการประเมินค่า Φ_p ตามสมการ (18) กับแผนการทดลอง X_0 เท่านั้น แผนการทดลองที่สร้างขึ้นหลังจากนี้ ใช้วิธีการดังต่อไปนี้

การประเมินค่า Φ_p แบบใหม่ หลังจากการแลกเปลี่ยนสมาชิกของแผนการทดลอง X ในแถวที่ i_1 และ i_2 ของคอลัมน์ k เมทริกซ์ D จะเปลี่ยนแปลงเพียงแถวที่ i_1 และ i_2 และคอลัมน์ i_1 และ i_2 เท่านั้น สำหรับทุกๆ $1 < j < n$ และ $j \neq i_1, i_2$ กำหนดให้

$$s(i_1, i_2, k, j) = |x_{i_2 k} - x_{jk}|^t - |x_{i_1 k} - x_{jk}|^t$$

ซึ่งเป็นสมการแทนระยะทางระหว่างจุดที่เปลี่ยนไปหลังจากการแลกเปลี่ยนสมาชิกแถวที่ i_1 และ i_2 ของคอลัมน์ k ในเมทริกซ์ D และให้

$$d'_{i_1 j} = d'_{j i_1} = [d_{i_1 j}^t - s(i_1, i_2, k, j)]^{1/t}$$

ในทำนองเดียวกัน

$$d'_{i_2 j} = d'_{j i_2} = [d_{i_2 j}^t - s(i_1, i_2, k, j)]^{1/t}$$

ซึ่ง $d'_{i_1 j}$ และ $d'_{i_2 j}$ คือระยะทางระหว่างจุดแทนที่จะถูกนำมาแทนที่ระยะทางระหว่างจุดเดิมใน D แล้วจึงคำนวณหาค่า Φ_p แบบใหม่ โดยใช้ระยะทางระหว่างจุดที่เปลี่ยนแปลงไปหลังการแลกเปลี่ยนสมาชิก กับค่า Φ_p เดิมก่อนการแลกเปลี่ยนสมาชิกดังสมการ (22)

$$\Phi'_p = \left[\begin{aligned} & \Phi_p^p + \sum_{j=1}^n [(d'_{i_1 j})^{-p} - (d_{i_1 j})^{-p}] \\ & + \sum_{j=1}^n [(d'_{i_2 j})^{-p} - (d_{i_2 j})^{-p}] \end{aligned} \right]^{\frac{1}{p}}$$

โดยที่ $j \neq i_1, i_2$ และ t เป็นจำนวนเต็มบวก

บทที่ 4

ผลการวิจัย

เนื้อหาในบทนี้ จะกล่าวถึงแผนการทดลองที่เหมาะสมที่สร้างจากอัลกอริทึมการสืบค้นที่นำเสนอมาทั้งหมด โดยจะแสดงการเปรียบเทียบประสิทธิภาพของอัลกอริทึมการสืบค้น โดยแบ่งการเปรียบเทียบเป็นสองส่วน คือ เปรียบเทียบในส่วนของคุณสมบัติของแผนการทดลอง (Accuracy) คือการพิจารณาค่า Φ_p ของแผนการทดลองที่เหมาะสมเมื่อสิ้นสุดการทำงานของอัลกอริทึม และเปรียบเทียบในส่วนของเวลาที่ใช้ในการทำงานของอัลกอริทึม (Efficiency) ภายใต้ขนาดของแผนการทดลองที่แตกต่างกัน

ผลการทดสอบและเปรียบเทียบ

การทดสอบประสิทธิภาพของอัลกอริทึมการสืบค้นที่นำเสนอ โดยใช้วิธีการเปรียบเทียบกับอัลกอริทึม MSA ที่นำเสนอโดย Thamma, et al. (2008) โดยการเปรียบเทียบแบ่งออกเป็น 2 ส่วนดังนี้

เปรียบเทียบคุณสมบัติของแผนการทดลองคือการพิจารณาค่า Φ_p ของแผนการทดลองที่เหมาะสมเมื่อสิ้นสุดการทำงานของอัลกอริทึม และใช้ค่าทางสถิติในการวิเคราะห์ข้อมูลคือค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน ค่าสูงสุด และค่าต่ำสุด และเปรียบเทียบการลดลงของค่า Φ_p เมื่ออัลกอริทึมมีจำนวนรอบการแลกเปลี่ยนสมาชิกที่เท่ากัน โดยกระทำกับแผนการทดลองขนาด $9 \times 2.51 \times 5$, 201×10 และ 801×20

เปรียบเทียบประสิทธิภาพของอัลกอริทึมโดยใช้เวลาการทำงานของอัลกอริทึมในการสร้างแผนการทดลองที่เหมาะสม และแสดงให้เห็นอัตราส่วนการลดลงของเวลาที่ใช้ในการทำงานของอัลกอริทึม คืออัตราส่วน MSA/ESE, MSA/MESE และ ESE/MESE

การเปรียบเทียบประสิทธิภาพทั้ง 2 ส่วนที่กล่าวมากระทำภายใต้ขนาดของแผนการทดลองที่แตกต่างกัน โดยแบ่งเป็น ปัญหาขนาดเล็ก (Small size design) ที่มีตัวแปรเข้าในปริภูมิการทดลองจำนวน 2 - 9 ตัวแปร และ ปัญหาขนาดใหญ่ (Large size design) ที่มีตัวแปรเข้าในปริภูมิการทดลองจำนวน 10 - 15 และ 20 ตัวแปร ซึ่งขนาดของแผนการทดลองมีผลกับการตั้งค่าพารามิเตอร์ของอัลกอริทึม ซึ่งส่งผลกระทบต่อจำนวนรอบการแลกเปลี่ยนสมาชิก และเวลาที่ใช้

1. ผลการทดสอบกับปัญหาขนาดเล็ก

แผนการทดลองที่มีตัวแปรเข้าจำนวน 2 - 9 ตัวแปร การตั้งค่าพารามิเตอร์สำหรับแต่ละอัลกอริทึมมีรายละเอียดดังนี้ MSA มีจำนวนรอบการทำงาน $i_{ms} = 1000$ สำหรับแผนการทดลองขนาด 2 - 6 ตัวแปร และ $i_{ms} = 500$ สำหรับแผนการทดลองขนาด 7 - 9 ตัวแปร (Thamma, et al. 2008) ส่วน ESE และ MESE ตั้งค่าให้มีจำนวนรอบการทำงานลูปนอกคือ 40 สำหรับแผนการทดลองขนาด 2 - 6 ตัวแปร และ 20 สำหรับแผนการทดลองขนาด 7 - 9 ตัวแปร สำหรับผลการทดสอบกับแผนการทดลองขนาดเล็ก เปรียบเสมือนการทดสอบวิธีการค้นหาแบบเชิงกลุ่มในกระบวนการค้นหาแบบสำรวจ (Exploration process) ของ ESE และ MESE เนื่องจากแผนการทดลองขนาดเล็ก การเพิ่มจำนวนรอบการทำงานลูปนอกให้กับ ESE และ MESE เป็นการเพิ่มเพื่อให้กระบวนการของ ESE และ MESE ให้สามารถหลุดออกการค้นหาจากแผนการทดลองที่เหมาะสมแบบเฉพาะที่จึงมีผลทำให้ค่า Φ_p ลดลงเพียงเล็กน้อยเท่านั้น

เปรียบเทียบคุณสมบัติของแผนการทดลอง

ตาราง 3 แสดงค่าสถิติพื้นฐานเช่น ค่าเฉลี่ย (Mean) ค่าต่ำสุด (Min) ค่าสูงสุด (Max) และค่าส่วนเบี่ยงเบนมาตรฐาน (S.D.) ของค่า Φ_p จากการทำงานซ้ำของแต่ละอัลกอริทึมจำนวน 10 ครั้ง ผลที่ได้คือในแผนการทดลองขนาด 9×2 อัลกอริทึม MSA สามารถสร้างแผนการทดลองที่มีคุณสมบัติดีกว่า ESE และ MESE โดยให้ค่า Φ_p เฉลี่ยเท่ากับ 4.273 ซึ่งต่ำกว่า MESE และ ESE และเมื่อพิจารณาค่าส่วนเบี่ยงเบนมาตรฐานที่เท่ากับ 0.000 แสดงให้เห็นว่าในการทำงานทั้ง 10 ครั้งของ MSA แผนการทดลองที่ได้หลังจากจบการทำงานให้ค่า $\Phi_p = 4.273$ ในทุกรอบ สำหรับแผนการทดลองขนาด 51×5 อัลกอริทึมที่สร้างแผนการทดลองที่มีคุณภาพดีที่สุดคือ ESE และ MSA โดยให้ค่า Φ_p โดยเฉลี่ยเท่ากันคือ 5.422 ซึ่งต่ำกว่า MESE ที่แผนการทดลองให้ค่า Φ_p โดยเฉลี่ยเท่ากันคือ 5.423 อยู่เล็กน้อย โดยทั้งสามอัลกอริทึมมีค่าส่วนเบี่ยงเบนมาตรฐานที่ต่างกันเพียงเล็กน้อย (MSA = 0.004, ESE = 0.006 และ MESE = 0.004) เท่านั้น และสำหรับแผนการทดลองขนาด 163×9 ซึ่งเป็นแผนการทดลองที่มีตัวแปรเข้ามากที่สุดในปัญหาขนาดเล็ก ESE ก็ยังสามารถสร้างแผนการทดลองที่มีคุณสมบัติที่ดีที่สุดโดยให้ค่า Φ_p โดยเฉลี่ยเท่ากับ 6.309 ซึ่งต่ำที่สุด รองลงมาคือ MESE ที่สามารถสร้างแผนการทดลองที่ให้ค่า $\Phi_p = 6.042$ ซึ่งต่ำกว่า MSA ที่ให้ค่า $\Phi_p = 6.045$ เล็กน้อย และทั้งสามอัลกอริทึมมีค่าส่วนเบี่ยงเบนมาตรฐานที่เท่ากันเมื่อพิจารณาที่ทศนิยม 3 ตำแหน่งคือ 0.001

สรุปเมื่อทำการพิจารณาโดยภาพรวมแล้วพบว่า ESE สามารถสร้างแผนการทดลองที่มีคุณสมบัติดีกว่า MSA และ MESE คือให้ค่า Φ_p โดยเฉลี่ยที่ต่ำกว่าทั้ง MSA และ MESE ที่สามารถสร้างแผนการทดลองที่ให้ค่า Φ_p ใกล้เคียงกัน และเมื่อพิจารณาค่าส่วนเบี่ยงเบนมาตรฐาน โดยภาพรวมแล้วพบว่าทั้ง MSA, ESE และ MESE มีการกระจายน้อย โดยเฉพาะเมื่อแผนการทดลองมีขนาดใหญ่ขึ้นซึ่งแสดงให้เห็นถึงการทำงานอย่างคงที่ (Consistency) ในแต่ละอัลกอริทึม

เปรียบเทียบประสิทธิภาพของอัลกอริทึม

ตาราง 4 เสนอผลการเปรียบเทียบประสิทธิภาพของ MSA, ESE และ MESE โดยพิจารณา เวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสม (Time) ในหน่วยวินาที (Second), จำนวนครั้งของการแลกเปลี่ยนสมาชิก (#Exchange) และอัตราส่วนเวลาการสร้างแผนการทดลองที่เหมาะสมระหว่าง MSA/ESE, MSA/MESE และ ESE/MESE ซึ่งผลที่ได้จากตาราง 4 แสดงให้เห็นว่า สำหรับแผนการทดลองขนาด 9×2 อัลกอริทึม MESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมโดยเฉลี่ยน้อยที่สุดคือ 2.296 วินาที โดยใช้เวลาน้อยกว่า ESE ที่ตามมาเป็นอันดับสองคือ 3.595 วินาที อยู่ประมาณ 1.565 เท่า และน้อยกว่า MSA ที่ใช้เวลา 19.993 วินาที และเมื่อพิจารณาที่จำนวนครั้งการแลกเปลี่ยนสมาชิก MESE ใช้จำนวนครั้งการแลกเปลี่ยนสมาชิกที่น้อยกว่า ESE อยู่เล็กน้อยคือ 5,415 และ 5,760 ตามลำดับซึ่งน้อยกว่า MSA ที่ใช้จำนวนครั้งการแลกเปลี่ยนสมาชิกไปถึง 47,140 ครั้งในการสร้างแผนการทดลองที่เหมาะสม และสำหรับแผนการทดลองขนาด 51×5 อัลกอริทึม MESE ก็สามารถสร้างแผนการทดลองที่เหมาะสมโดยใช้เวลาและจำนวนครั้งการแลกเปลี่ยนสมาชิกโดยเฉลี่ยน้อยที่สุดคือ 313.067 วินาที ในการแลกเปลี่ยนสมาชิก 118,950 ครั้ง ซึ่งน้อยกว่า ESE ที่ใช้เวลาเท่ากับ 550.762 วินาที ในการแลกเปลี่ยนสมาชิก 150,000 ครั้ง และ MSA ที่ใช้เวลา 751.540 วินาที ในการแลกเปลี่ยนสมาชิก 212,889 ครั้ง ตามลำดับ

ตาราง 3 แสดง Φ_p ในเชิงสถิติที่ได้จาก MSA, ESE และ MESE ของปัญหาขนาดเล็ก

LHDs	Algorithm	$(p = 5, t = 2)$			
		Min	Max	Mean	SD
9 × 2	MSA	4.273	4.273	4.273	0
	ESE	4.273	4.344	4.287	0.029
	MESE	4.273	4.344	4.280	0.022
19 × 3	MSA	4.898	4.938	4.917	0.015
	ESE	4.895	4.934	4.913	0.013
	MESE	4.898	4.934	4.917	0.014
33 × 4	MSA	5.204	5.214	5.208	0.003
	ESE	5.195	5.216	5.206	0.006
	MESE	5.205	5.214	5.208	0.003
51 × 5	MSA	5.417	5.43	5.422	0.004
	ESE	5.415	5.431	5.422	0.006
	MESE	5.418	5.43	5.423	0.004
73 × 6	MSA	5.585	5.597	5.590	0.003
	ESE	5.588	5.595	5.590	0.002
	MESE	5.588	5.597	5.591	0.002
99 × 7	MSA	5.75	5.757	5.754	0.002
	ESE	5.752	5.757	5.754	0.001
	MESE	5.755	5.762	5.759	0.002
129 × 8	MSA	5.9	5.904	5.902	0.001
	ESE	5.898	5.901	5.900	0
	MESE	5.902	5.906	5.904	0.001
163 × 9	MSA	6.043	6.047	6.045	0.001
	ESE	6.038	6.042	6.039	0.001
	MESE	6.039	6.044	6.042	0.001

ตาราง 4 แสดงประสิทธิภาพของ MSA, ESE และ MESE ของปัญหาขนาดเล็ก

LHDs	Algorithm	Performance (Average)		Time ratio		
		Time (sec.)	#Exchange	MSA/ESE	MSA/ESEE	ESE/MESE
2 × 9	MSA	19.993	47140			
	ESE	3.595	5760	5.561	8.707	1.565
	MESE	2.296	5415			
19 × 3	MSA	123.441	133418			
	ESE	67.74	41040	1.822	3.601	1.976
	MESE	34.275	37121			
33 × 4	MSA	351.293	212889			
	ESE	344.918	126720	1.018	2.129	2.090
	MESE	164.972	98580			
51 × 5	MSA	751.54	284931			
	ESE	550.762	150000	1.364	2.400	1.759
	MESE	313.067	118950			
73 × 6	MSA	1352.469	351624			
	ESE	734.981	150000	1.840	2.640	1.434
	MESE	512.225	130500			
99 × 7	MSA	1056.969	194549			
	ESE	649.427	100000	1.627	2.879	1.769
	MESE	367.115	67785			
129 × 8	MSA	1496.892	204924			
	ESE	835.196	100000	1.792	2.585	1.442
	MESE	579.057	79310			
163 × 9	MSA	1987.253	207458			
	ESE	1084.572	100000	1.832	2.278	1.243
	MESE	872.018	88955			

สรุปสำหรับทุกขนาดของของแผนการทดลองในปัญหาขนาดเล็ก อัลกอริทึม ESE และ MESE ใช้จำนวนครั้งของการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองน้อยกว่า MSA เกิน 50% เกือบทุกขนาดของปัญหา ซึ่งส่งผลให้ใช้เวลาน้อยกว่าด้วย ซึ่งให้เห็นว่า ESE และ MESE มีประสิทธิภาพเหนือกว่า MSA ในภาพรวม และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ MESE พบว่า MESE ใช้จำนวนครั้งในการแลกเปลี่ยนสมาชิกน้อยกว่า ESE ส่งผลให้ใช้เวลาน้อยกว่าโดยเฉลี่ยประมาณ 1.65 เท่าในทุกขนาดของแผนการทดลอง

การพิจารณาเปรียบเทียบการลดลงของค่า Φ_p ของแผนการทดลองขนาด 9×2 โดยให้แต่ละอัลกอริทึมทำการแลกเปลี่ยนสมาชิกเพื่อสร้างแผนการทดลองใหม่จำนวน 4,320 รอบ จากแผนการทดลองเริ่มต้นที่ต่างกันอย่างการสุ่ม กระทำทั้งหมด 10 ครั้งแล้วทำการหาค่าเฉลี่ยของจำนวนการครั้งแลกเปลี่ยนสมาชิก เพื่อพิจารณาการลดลงของค่า Φ_p ซึ่งเป็นการแสดงให้เห็นว่าแต่ละอัลกอริทึมสามารถสร้างแผนการทดลองที่เข้าใกล้แผนการทดลองที่เหมาะสม



บทที่ 5

สรุปผลการวิจัย

การวิจัยครั้งนี้มีจุดมุ่งหมายเพื่อนำเสนออัลกอริทึมการสืบค้นที่มีประสิทธิภาพสำหรับสร้างแผนการทดลองที่เหมาะสมเพื่อการจำลองการทดลองด้วยคอมพิวเตอร์โดยเน้นการปรับปรุงประสิทธิภาพของ Simulate annealing algorithm และ Stochastic evolutionary algorithm จากนั้นวัดประสิทธิภาพของวิธีการที่นำเสนอโดยพิจารณาคุณสมบัติของแผนการทดลองที่สร้างได้ รวมไปถึงเวลาที่ใช้ในการสืบค้น เพื่อหาข้อสรุปและข้อเสนอแนะในการใช้งานของอัลกอริทึมต่าง ๆ

สรุปผลการวิจัย

จากการวิจัยในครั้งนี้ อัลกอริทึมการสืบค้นที่แบบสโตนแคสติกเชิงวิวัฒนาการที่พัฒนาขึ้นสำหรับการสร้างแผนการทดลองที่เหมาะสม โดยการนำข้อดีจากอัลกอริทึม MSA และ ESE มารวมเข้าด้วยกัน จากผลการวิจัยสามารถสรุปได้ดังนี้

1. ESE ที่ใช้หลักการแบ่งการทำงานเป็น 2 ลูป คือ ลูปใน และลูปนอกสามารถทำงานได้อย่างมีประสิทธิภาพ และทำงานได้ดีกว่า MSA เพราะการทำงานของลูปนอกทำหน้าที่ปรับวิธีการค้นหาให้เหมาะสมกับแผนการทดลองปัจจุบันในขณะนั้น ไม่ได้เน้นไปที่การปรับแผนการทดลองที่ให้ดีขึ้นเรื่อยๆ ในทิศทางเดียวเหมือนกับ MSA

2. MESE มีประสิทธิภาพสูงกว่า MSA เมื่อพิจารณาที่จำนวนการแลกเปลี่ยนสมาชิก และเวลาที่ใช้ในการทำงานในหน่วยวินาที และ MESE สามารถสร้างแผนการทดลองที่ให้ค่า Φ_{avg} น้อยกว่า MSA กับปัญหาขนาดใหญ่

3. MESE มีประสิทธิภาพสูงกว่า ESE เมื่อพิจารณาที่จำนวนการแลกเปลี่ยนสมาชิกในปัญหาขนาดเล็ก แต่ในปัญหาขนาดใหญ่ MESE ไม่สามารถสร้างแผนการทดลองที่เหมาะสมได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยกว่า ESE มากนัก สืบเนื่องมาจากกฎการหยุดสืบค้นไม่สามารถทำให้ MESE หยุดทำงานได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยลง ซึ่งจากการศึกษาพบว่า เมื่อปัญหามีขนาดใหญ่แผนการทดลองสุดท้ายที่ได้ในแต่ละรอบการทำงานจะให้ค่า Φ_{avg} ที่น้อยลงเสมอในทุกๆ รอบ สรุปคือ MESE มีประสิทธิภาพสูงกว่า ESE เล็กน้อยเมื่อพิจารณาเปรียบเทียบเวลาในการสร้างแผนการทดลองทั้งกับปัญหาขนาดเล็กและปัญหาขนาดใหญ่

4. คุณสมบัติของแผนการทดลองที่วัดโดยใช้เกณฑ์การเลือกค่าเหมาะสม Φ_{avg} โดยเฉลี่ยแล้ว MSA, ESE และ MESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ใกล้เคียงกันในปัญหาขนาดเล็ก แต่กับปัญหาขนาดใหญ่ ESE และ MESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ดีกว่า MSA

ข้อเสนอแนะ

1. ควรเลือกกลุ่มการออกแบบให้หลากหลายมากขึ้นเช่น Symmetric Latin hypercube design (SLHD) หรือ Orthogonal Latin hypercube design (OLHD)

2. ควรพัฒนากฎการหยุดสืบค้นให้เหมาะสมมากขึ้น โดยการประยุกต์ใช้วิธีการทางสถิติ เพื่อไม่ให้อัลกอริทึมเสียจำนวนรอบการทำงานโดยไม่มีประโยชน์

3. ควรนำแผนการทดลองที่เหมาะสมที่ได้จากการสร้างของ MESE และ ESE ไปพัฒนาโมเดลเพื่อการพยากรณ์ เพื่อวัดประสิทธิภาพของโมเดลที่ได้จากแผนการทดลองว่ามีความแม่นยำเพียงใด

บรรณานุกรม

1. Allen, T.T., Bernshteyn, M.A., Kabiri-Bamoradian, K., 2003. Constructing Meta-Models for Computer experiments. *Journal of Quality Technology* 35, 264-274.
2. Box, G.E.P, Hunter, W.G., Hunter, J.S., 2005. *Statistics for Experimenters: Design, Innovation, and Discovery*. 2nd Edition, John Wiley & Sons, New York.
3. Butler, N.A., 2001. Optimal and orthogonal latin hypercube designs for computer experiments. *Biometrika*, 88(3): 847-857.
4. Castillo, E., 1988. *Extreme Value Theory in Engineering*. Academic Press, INC, San Diego.
5. Cioppa, T.M., Lucas, T.W. 2007. Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes. *Technometrics*, 49(1): 45-55
6. Crary, S.B. 2002. Design of Computer Experiments for Metamodel Generation. *Analog Integrated Circuits and Signal Processing*, 32, 7-16, 2002.
7. Cressie, N. A. C. 1991. *Statistics for Spatial Data*. John Wiley, New York.
8. Cheong, Y. N., 2001. *Heuristic Algorithms for Routing Problems*. PhD thesis, Curtin University of Technology.
9. Etman, L.E.P 1994. Design and analysis of computer experiments: The method of Sacks et al., Engineering Mechanics report WFW 94.098, Eindhoven University of Technology.
10. Fang, H., Horstemeyer, M.F., 2006. Global response approximation with radial basis functions. *Engineering Optimization*, 38(4): 407-424.
11. Fang, K. T., Li, R., Sudjianto, A. 2005. *Design and modeling for computer experiments*. Chapman & Hall/CRC, London UK.
12. Grosso, A., Jumali, A.R.M.J.U., Locatelli, M. 2009. Finding maximin latin hypercube designs by Iterated Local Search heuristics. *European Journal of Operational Research*, 197: 541-547.
13. Hock, W., Schittkowski, K., 1981. *Test examples for nonlinear programming codes*. Berlin, Heidelberg, New York: Springer.
14. Hussian, M.F., Barton, R. R., Joshi, S.B. 2002. Metamodeling: Radial basis functions, versus polynomials. *European Journal of Operational Research*, 138: 142-154.
15. Jin, R., Chen, W., and Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134: 268-287.
16. Johnson, M. E., Moore, I.M., and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26:131-148.
17. Joseph, V.R., Hung, Y., 2008. Orthogonal-Maximin Latin Hypercube Designs. *Statistica Sinica*, 18: 171-186.
18. Koehler, J., Owen, A.B., 1996. Computer experiments. *Handbook of Statistics*, Elsevier Science, New York, pp. 261-308.
19. Leary, S., Bhaskar, A., and Keane, A. 2003. Optimal orthogonal-array-based latin hypercubes. *Journal of Applied Statistics*, 30(5): 585-598.

20. Li, W. and Wu, C.F.J. (1997). Columnwise-pairwise algorithms with applications to the construction of supersaturated designs. *Technometrics*, 39:171-179.
21. Liefvandahl, M. and Stocki, R. (2006). Study on algorithms for optimization of latin hypercubes. *Journal of Statistical Planning and Inference*, 136:3231-3247.
22. McDonald, D.B., Grantham, W.J., Tabor, W.L., Murphy, M.J., 2007. Global and local optimization using radial basis function response surface models. *Applied Mathematics Modelling*, 31: 2095-2110.
23. Mckay, M., Beckman, R., Conover, W., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21, 239-246.
24. Montgomery, D.C., 2009. *Design and analysis of Experiments*. John Wiley & Sons, New York.
25. Morris, M.D., Mitchell, T.J., 1995. Exploratory design for computer experiments. *Journal of Statistical planning and inference*, 43: 381-402.
26. Mullur, A.A., Messac, A., 2006. Metamodeling using extended radial basis functions: a comparative approach. *Engineering with Computers*, 21: 203-217.
27. Na-Udom, A., 2007. Experimental design methodology for modeling response from computer simulated experiments. Ph.D. thesis, Curtin University of Technology.
28. Na-udom, A., 2007. Comparison of designs for computer simulated experiments. Annual meeting on Statistics 2007, 24-25 May 2007, Chaum, Phetchaburi, Thailand.
29. Prescott, P., 2009. Orthogonal-column Latin hypercube design with small samples. *Computational Statistics and Data Analysis*, 53: 1191-1200.
30. Rungrattanaubol, J., Na-udom, A., 2007. Comparison of evolutionary search algorithms in computer simulated experiments. The 11th National Computer Science and Engineering Conference (NCSEC 2007), 19-21 November 2007, Miracle Grand Hotel, Bangkok, Thailand.
31. Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P., 1989. Design and Analysis of Computer Experiments. *Statistical Science* 4(4), 409-435.
32. Shewry, M. and Wynn, H. P., 1987. Maximin entropy design. *Journal of applied statistics*, 14(2):165-170.
33. Simpson, T.W., Lin, D. K. J. and Chen, W. 2001. Sampling strategies for computer experiments: Design and analysis. *International Journal of Reliability and Applications*, 2(3): 209-240.
34. Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K., 2001. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17(2):129-150.
35. Steinberg, D.M., 2006. A construction method for orthogonal Latin hypercube designs. *Biometrika*, 93(2): 279-288.
36. Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P., Mitchell, T.J., and Morris, M.D., 1992. Screening, predicting, and computer experiments. *Technometrics* 34, 15-25.

37. Ye, K. Q., 1998. Orthogonal column latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association* 93, 1430-1439.
38. Ye, K.Q., Li, W., Sudjianto, A., 2000. Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of Statistical planning and inference* 90, 145-159.



ภาคผนวก

1. จดหมายตอบรับการตีพิมพ์บทความวิจัยลงในการประชุมวิชาการระดับนานาชาติ Applied Statistics
2. Manuscript ที่ได้รับการยอมรับให้ตีพิมพ์ในรายงานสืบเนื่องจากการประชุมวิชาการระดับนานาชาติ





King Mongkut's University of Technology Thonburi

126 Pracha-utid Rd., Bangmod, Toong-kru, Bangkok 10140 Thailand Tel. (662) 470-8000, 427-0039 Fax. (662) 427-9860

April 11, 2013

Dear Author,

Your manuscript entitled "A study on search algorithms for constructing optimal designs" has been carefully reviewed by three ICAS 2013 academic peer reviewers. On behalf of ICAS 2013 academic committee, it is my pleasure to inform you that your paper has been accepted for oral presentation in ICAS 2013. Your full paper will also be published in the conference proceedings.

Please register online no later than April 20, 2013 for early registration rate. The presentation schedule will be announced online. You are required to give a presentation at the designated time.

I look forward to seeing you in ICAS 2013.

Sincerely yours,

A handwritten signature in black ink, appearing to read 'Adisak', is written over a large, faint watermark of the university's seal. The seal depicts an elephant and contains Thai text.

Assoc.Prof.Adisak Pongpullponsak

EDITOR : International Conference on Applied Statistics 2013 (ICAS 2013)

E-mail: adisak.pon@kmutt.ac.th

<http://icas2013.kmutt.ac.th>

A Study on Search Algorithms for Constructing Optimal Designs

Jaratsri Rungrattanaubol*

Department of Computer Science and Information
Technology, Faculty of Science
Naresuan University
Phitsanulok, Thailand

*Corresponding author: jaratsrir@nu.ac.th

Anamai Na-udom

Department of Mathematics, Faculty of Science,
Naresuan University
Phitsanulok, Thailand
anamain@nu.ac.th

Abstract: Computer simulated experiments (CSE) are often used in science and engineering applications. The nature of CSE is that time consuming and computationally expensive to run. Normally, the output response from computer simulated experiments is deterministic. Consequently the space filling designs, which focus on spreading design points over a design space, are necessary. Latin hypercube designs (LHD) are normally practiced in the context of CSE. The optimal LHD for a given dimension of problem is constructed by using a search algorithm under a pre-specified optimality criterion. Usually this searching process takes a long time to terminate, especially when the dimension of the problem is large. This paper proposes the methods to enhance the performance of search algorithms which are widely used in the context of CSE. The comparative studies are employed based on a range of problems and optimality criteria. The results indicate that the proposed method can improve the capability of the search algorithms for constructing the optimal LHD.

Keywords: Computer simulated experiments; Latin hypercube designs; simulated annealing algorithms; enhanced stochastic evolutionary algorithm; optimality criteria

I. INTRODUCTION

Recently computer simulated experiments (CSE) have replaced classical experiments to investigate a physical complex phenomena, especially when classical (physical) experiments are not feasible. For example, the use of reservoir simulator to predict ultimate recovery of oil, the use of finite element codes to predict behavior of metal structure under stress, and so on [6]. The nature of computer simulated experiments is deterministic [16, 19]; hence identical settings of input variables always produce an identical set of output response. Therefore, space filling designs that aim to spread the design points over a region of interest are necessary. The most popular class of space filling design in the context of computer simulated

experiments is Latin hypercube design (LHD). LHD design was originally proposed by McKay and co-workers [11] in 1979. The ultimate goal of selecting the settings of input variables is to attain the coverage of all design regions of interest.

As mentioned before that the space filling designs are preferred in the context of computer simulated experiments. Space filling designs or the optimal LHD can be constructed through combinatorial methods (non-search algorithm) [2, 20] or searching for a design through search algorithms [12, 13]. The former method generates design with good design properties but it is restricted in terms of a design size. For example methods proposed by Butler [2] are limited to a design size of a prime number. The latter method is based largely on improving design by exchanging between the pairs of design points. Exchange algorithms can be time consuming to implement, however, the generated design are flexible and straightforward. The CSEs are usually complex and consist of many input variables to investigate [1]. In this case a large number of runs are required to estimate the parameters corresponding to the factors of interest in the model. For example, if the problem of interest consists of d input variable and n number of runs, the total number of LHD is $(n!)^d$. Obviously this number explodes exponentially as the values of n and d increase; hence the full space of LHD cannot be explored. In this case we need the search algorithms to lead us to a good design with respect to an optimality criterion. The key idea of all existing search algorithms is to use some kinds of exchange procedures to move towards the better designs.

The search based approach for selecting a design is implemented by combining search algorithms and the optimality criterion [13]. For example, Morris and Mitchell [12] adopted a version of Simulated Annealing algorithms (SA) to search for optimal LHDs with respect to ϕ_p criterion. Li and Wu [8] proposed a columnwise-pairwise

algorithm (CP) with respect to the D efficiency criterion. It was reported that CP is very simple and easy to implement. The only parameter required to set as a priori is the tolerance level (tl). Further, CP is able to generate a good supersaturated design and it can be used along with various optimality criteria [12]. In order to avoid the problem of convergence and the search being stuck at a local optimum value, usually multiple search with different starting points are performed. The best result, among different trials, is selected as optimal design. It should be noted that for large dimensional problems, CP algorithm can be time consuming to implement. Ye and his co-workers [20] adapted CP algorithm to search for symmetric LHD under various optimality criteria such as entropy and ϕ_p criteria. Park [13] proposed a row-wise element exchange algorithm along with IMSE and entropy criteria. Leary et al [7] adapted CP and SA algorithms to construct the optimal designs within the orthogonal-array based Latin hypercube class by using the ϕ_p criteria. Jin et al. [4] developed an enhanced stochastic evolutionary algorithm (ESE) to search for the best design considering various optimality criteria such as a maximin distance criterion, ϕ_p criterion and entropy criterion. ESE has received wide attention from researchers due to its performance in constructing the optimal LHD. Liefvendahl and Stocki [10] applied a version of Genetic algorithm (GA) to search for the optimal LHD considering ϕ_p and a maximin distance criterion. A similar work can be found in [9] as the authors applied GA for constructing maximin designs. Grosso et al. [3] used the iterated local search algorithm and SA in constructing the optimal LHD under maximin distance and ϕ_p criterion. Vianna et al. [18] proposed the algorithm for fast optimal LHD by using the idea of seed design under maximin distance and ϕ_p criterion. Due to the popularity of SA and ESE along with ϕ_p criteria, this paper presents the efficient method to improve the capability of SA and ESE under ϕ_p criterion. In the following sections we present details of these search algorithms, followed by the details of the optimality criteria. The enhancement methods on SA and ESE are also presented in section III. The results of the enhancement methods will be presented in the result section and conclusion will be given in section V respectively.

II. EXPERIMENTAL DESIGN AND OPTIMALITY CRITERION

This section presents the details of LHD and the steps of search algorithms including the enhancement methods to improve their performance in constructing the optimal LHD.

A. Latin hypercube design (LHD)

LHD can be constructed based on the idea of stratified sampling [11] to ensure that all subregions in the divided input variable space will be sampled with equally probability. A Latin hypercube sampling has

$$X_{ij} = \frac{\pi_{ij} - U_{ij}}{n} \tag{1}$$

, where π_{ij} are the elements of an $n \times d$ matrix comprising of columns $\pi_j (j=1,2,K,d)$. Each column $\pi_j (j=1,2,K,d)$ is independent random permutation of number 1 through n and U_{ij} are $n \times d$ values of independent $U[0,1]$ random variables independent of the π_{ij} . The example of LHD is shown in Figure 1.

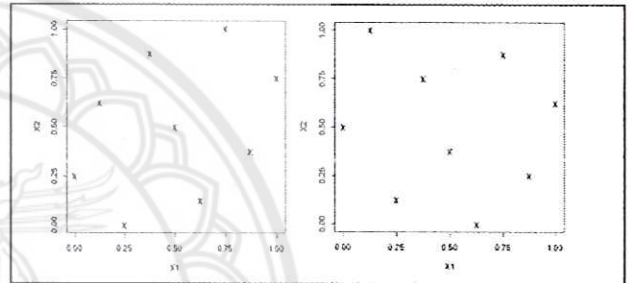


Figure 1. Example of 9x2 LHD

B. Element exchange operation

The element exchange operation to construct a new LHD design is developed by using the concept of column-wise operation proposed by Li and Wu [8]. The process is randomly interchange two distinct elements in a randomly selected column as shown in Figure 2. After an element exchange has been performed, the LHD properties still remains.

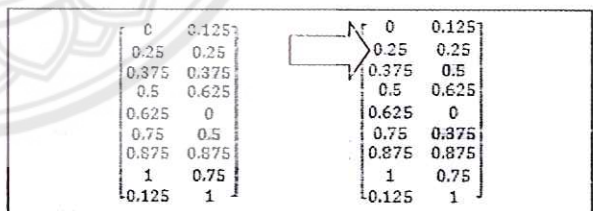


Figure 2. Element exchange in the 2nd column of a 9x2 LHD

C. The ϕ_p optimality criterion

Morris and Mitchell [12] proposed a modification class of maximin distance criterion to search for the optimal design.

For a given design X , the Euclidean intersite distance between any two design points can be calculated from

$$d(x_i, x_j) = \left[\sum_{k=1}^d (x_{ik} - x_{jk})^2 \right]^{1/2} \quad (2)$$

By using (2), all intersite distances for every pairs of design points are calculated and can be expressed in the symmetric matrix form as follows.

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1j} \\ d_{21} & d_{22} & \dots & d_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nj} \end{bmatrix}$$

Let a Euclidean distance list (d_1, d_2, \dots, d_m) be the distinct elements list from the smallest to largest. Also define index list (J_1, J_2, \dots, J_m) , which J_j is the number of pairs of sites in the design separated by distance d_j . Thus X is a maximin design if among available designs, it maximizes d_j while J_j is minimized. The scalar criterion can be expressed as

$$\phi_p = \left[\sum_{j=1}^m J_j d_j^{-p} \right] \quad (3)$$

where p is a positive integer, J_j and d_j specified from X . The design that minimizes ϕ_p is a maximin LHD in the class. In this study, the adaptive form of ϕ_p [4] which is simpler than (3) to implement is considered

$$\phi_p = \left[\sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{d_{ij}^p} \right]^{1/p} \quad (4)$$

After ϕ_p value has been calculated, a design that minimizes ϕ_p is considered as an optimal design in the class.

III. SEARCH ALGORITHMS

This section presents the details of search algorithms used in this study including the enhancement methods to improve the performance of the search algorithms.

A. Modification of simulated annealing algorithm (MSA)

Morris and Mitchell [12] developed a simulated annealing algorithm to search for an optimal LHD using ϕ_p optimality criterion. The design that minimizes ϕ_p value is considered as the best design in the class. The steps of SA are presented as follows.

<p>Step 1: Set initial values I_{\max} (maximum number of perturbation to seek improvement) t_0 (initial cooling temperature) C_t (factor by which t_0 is reduced when no improvement in ϕ_p) n (Number of runs) d (Number of input variables)</p> <p>Step 2: Generate a random LHD, X of given order $n \times d$. Let $X_{best} = X, t = t_0$</p> <p>Step 3: Set $I = 1, Label = 0$</p> <p>Step 4: Let $X_{try} = X$</p> <p>Randomly select a column say j, of matrix X_{try} and exchange two randomly selected elements of column j, say $X_{ij} \leftrightarrow X_{lj}$</p> <p>Step 5: Set $X = X_{try}, Label = 1$, If $\phi_p(X) - \phi_p(X_{try}) \geq \epsilon t$ or with probability $e^{-[\phi_p(X_{try}) - \phi_p(X)]/t}$</p> <p>Step 6: If $\phi_p(X_{try}) < \phi_p(X_{best})$, set $I = 1$ and $X_{best} = X_{try}$, else $I = I + 1$.</p> <p>Step 7: If $I < I_{\max}$ go to Step 4.</p> <p>Step 8: If $Label = 1$, set $t = t \times C_t$, go to Step 3.</p> <p>Step 9: Stop and report X_{best}.</p>
--

SA requires parameter settings, t_0 , I_{\max} , FAC_t and p . In this study, we use the heuristic methods to find the best set of parameters for use in SA. The choice of initial parameters for SA can be found in [12]. It was also reported in the paper that SA performed very well in terms of moving away from the local optimum value of ϕ_p criterion.

The emphasis of this paper is on the modification of SA by applying the calculation of ϕ_p criterion by using the method that avoids re-calculating of ϕ_p value. As mentioned before, SA uses the exchange procedure between two pairs of points within the randomly selected column. Hence, after an exchange between rows i_1 and i_2 within column k ($x_{i_1 k} \leftrightarrow x_{i_2 k}$), only elements in rows i_1 and i_2 , and columns i_1 and i_2 are changed in the distance matrix D [4].

For any $1 \leq j \leq n$ and $j \neq i_1, i_2$ let:

$$s(i_1, i_2, k, j) = |x_{i_1 k} - x_{j k}|^r - |x_{i_2 k} - x_{j k}|^r \quad (5)$$

then

$$d'_{i_1 j} = d'_{j i_1} = [d'_{i_1 j} + s(i_1, i_2, k, j)]^{1/r} \quad (6)$$

and

$$d'_{i_2 j} = d'_{j i_2} = [d'_{i_2 j} + s(i_1, i_2, k, j)]^{1/r} \quad (7)$$

Thus new ϕ_p is computed by

$$\phi'_p = \left[\frac{\phi_p^p + \sum_{1 \leq j \leq n, j \neq i_1, i_2} [(d'_{i_1 j})^{-p} - (d_{i_1 j})^{-p}] + \sum_{1 \leq j \leq n, j \neq i_1, i_2} [(d'_{i_2 j})^{-p} - (d_{i_2 j})^{-p}]}{2} \right]^{1/p} \quad (8)$$

As shown in (5) to (8), only some rows and columns are updated to calculate ϕ_p criterion in MSA. Hence the complexities or Big O of MSA is much smaller than SA as presented in Table 1.

TABLE 1. The complexities to calculate ϕ_p criterion in SA and MSA

BigO(SA)	$O(dn^2) + O(n^2 \log_2(p))$
BigO(MSA)	$O(n) + O(n \log_2(p))$

B. ESE and modification of ESE algorithm (MESE)

As presented in the previous section, the complexity of MSA is less than the original SA. Hence MSA is recommended for use in constructing the optimal design for CSE if time constraint is of interest. Jin et al. [4] proposed a new algorithm called enhanced stochastic evolutionary (ESE) algorithm and did a comparison between ESE and the existing algorithms such as CP and SA. The results showed that ESE is superior over the other algorithms in terms of computational time burden and the number of exchanges required for generating the optimal LHD design. According to the goodness of MSA and ESE, we combine them together to improve the search process. In next section we present the steps of ESE including the methods to improve the performance of ESE.

ESE was developed from stochastic evolutionary (SE) algorithm proposed by Sabb and Rao [15]. It contains 2 nested loop called inner and outer loops. The inner loop performs a local search process by constructing a new design and decides whether to accept new design or not. In the inner loop, both of acceptance ratio and improvement ratio are recorded. The outer loop works as a controller of the inner loop as it performs a global search by adjusting the threshold (Th) based on acceptance ratio and improvement ratio from the inner loop. The steps of ESE are presented as follows.

- Step 1:** Set initial parameters and design X_0 in the outer loop, $X = X_0$, $i = 0$, $n_{act} = 0$ and $n_{imp} = 0$
- Step 2:** Construct a set of new design X_{try}
- Step 3:** Select the best design X_{try} from this set
- Step 4:** Decide to accept the best design X_{try} and replace the current best design X from as shown in Figure 3.
- Step 5:** If X_{try} is better than the global best design X_{best} , replace it with X_{try} and increase n_{imp} by 1 ($n_{imp} = n_{imp} + 1$).
- Step 6:** Terminate the inner loop if $i > M$, else go to step 2.

The flowchart of the inner loop for ESE is visualized in Figure 3.

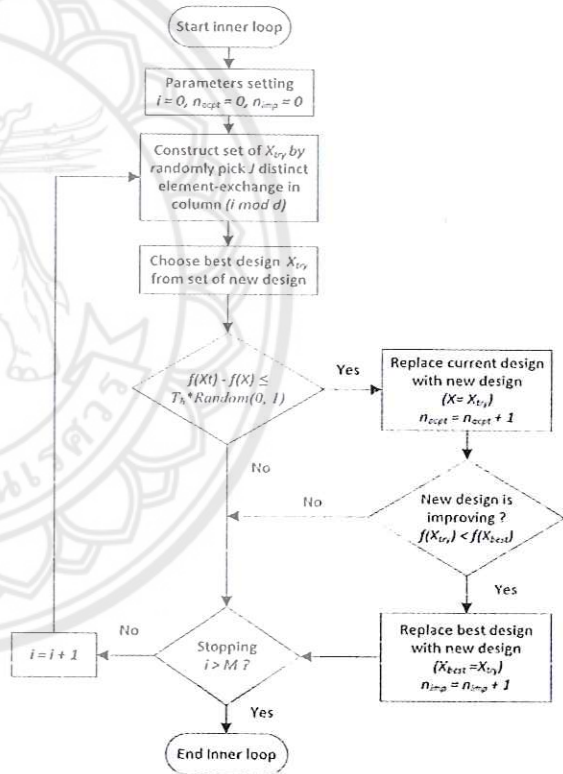


Figure 3. Flowchart of ESE inner loop [6]

In this study the parameters J is set to be $\binom{n}{2}/5$ but no larger than 50, and the parameter M is in a range of $2 \times \binom{n}{2} \times d/J \leq M \leq 100$. The outer loop of ESE is presented in Figure 4.

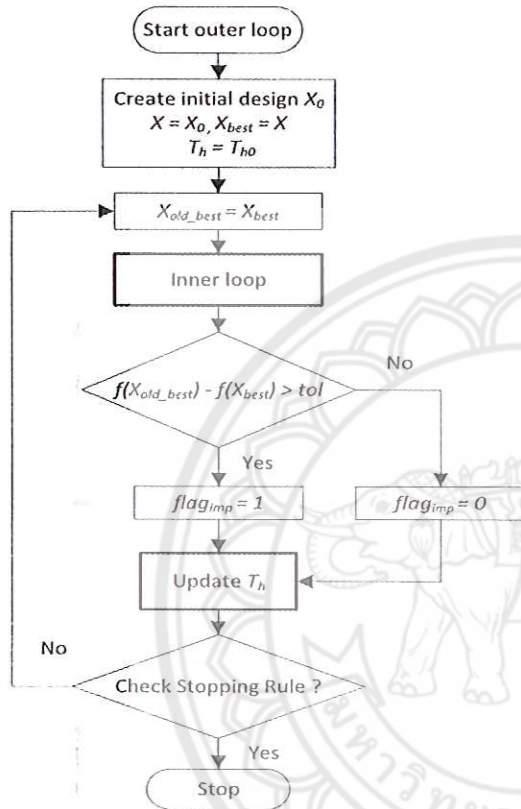


Figure 4. Flowchart of ESE outer loop [4]

The details of outer loop are given below.

- Step 1:** Randomly generate an initial design X_0 and set $X = X_0$, $X_{best} = X$, initialize $T_h = 0.005 \times \phi_p(X_0)$ and $T_h = T_h$.
- Step 2:** Set $X_{old_best} = X_{best}$.
- Step 3:** Go to the inner loop process.
- Step 4:** Select a method to update T_h , by setting $flag_{imp}$.
- Step 5:** Update T_h (discussed later for more details).

Step 6: Terminate the search by using a stopping rule, else go to step 2.

The tolerance level (tl) is set to 0.0001 as it was observed from the empirical study that the smaller value does not improve the search process. The process of updating the value of T_h in step 5 is divided into 2 processes called improving process and exploration process, respectively. The search process works as the improving process when $flag_{imp} = 1$, if the best design X_{best} is improved in the inner loop. If not, the search process will be in the exploration process ($flag_{imp} = 0$). In improving process ($flag_{imp} = 1$), T_h is adjusted in order to find the local best LHD based on an acceptance ratio (n_{act}/p) and improvement ratio (n_{imp}/M). If $n_{act}/M > \beta_1$ and $n_{imp}/M < n_{act}/M$, then T_h is decreased by $T_h = \alpha_1 \times T_h$. If $n_{act}/M > \beta_1$ and $n_{imp}/M = n_{act}/M$, then T_h is unchanged. Otherwise, T_h is increased by $T_h = T_h / \alpha_1$, where $0 < \alpha_1 < 1$ and $0 < \beta_1 < 1$, we use $\alpha_1 = 0.8$ and $\beta_1 = 0.1$ as suggested by Jin et al. [4]. Further the results obtained from our empirical studies also indicate that β_1 should be set to a small value. In the exploration process ($flag_{imp} = 0$), T_h will be adjusted to drive the algorithm to move far away from a local optimal design based on the range of accept ratio. If $n_{act}/M < \beta_2$, then T_h is increased until $n_{act}/M > \beta_3$ by equation $T_h = T_h / \alpha_3$. If $n_{act}/M > \beta_3$, then T_h is decreased till $n_{act}/M < \beta_2$ by equation $T_h = T_h \times \alpha_2$, where $0 < \alpha_2 < \alpha_3 < 1$ and $0 < \beta_2 < \beta_3 < 1$, we set $\alpha_2 = 0.9$, $\alpha_3 = 0.7$. While β_2 should be small, we set $\beta_2 = 0.1$ and β_3 should be large enough so we set $\beta_3 = 0.8$, as recommended in [4].

C. Modification of ESE (MESE)

This section we present the enhancement method on ESE. The modified version is called MESE. We combine the advantage of SA (i.e. local search process) and the advantage of ESE (i.e. global search process) together to improve the search process. MESE contains 2 nested loops as displayed in Figure 5. The outer loop is similar to the ESE except that there is only one change in a stopping rule as in step 6. The maximum number of cycles used is replaced by the following condition. If a local best design after the inner loop X_{best} is not improved from the global

best design ($X_{globalbest}$) for δ consecutive times, then the search process will be terminated. In this study we set $\delta = 10$.

The major enhancement was made in the inner loop. There are many changes have been made in step 2, step 5 and step 6. In step 2, the process for constructing a new design X_{try} is changed to element-exchange in column ($i \bmod d$) for all J iterations while the original ESE used the random strategy to pick J distinct element-exchange in column ($i \bmod d$). By doing this, the computational complexity decreases from $O(n^2)$ to $O(n)$. As can be seen in ESE process, a random element exchange for all J iteration is required in all i^{th} iterations, so all distinct $i-1$ loops must be checked. Hence the complexity is $O(n(n-1)) = O(n^2)$. In MESE, we adapt the process of element-exchange from SA shown in Figure 6.

So in any J iteration, element exchange of a current design X in column $i \bmod d$ is independent. Thus there is no need to perform all J iterations. It is obvious that the computation complexity decreases to $O(n)$. In step 5, if a new design X_{try} is improved (better than the best achieved design, X_{best}), let $j = 0$ otherwise increase j by 1 ($j = j + 1$). Finally, in step 6 of the inner loop, a stopping rule is modified to if $i > M$ or $j > C_{max}$. In this study, we set $C_{max} = 10$. All simulation studies presented in this paper were performed using R program [14].

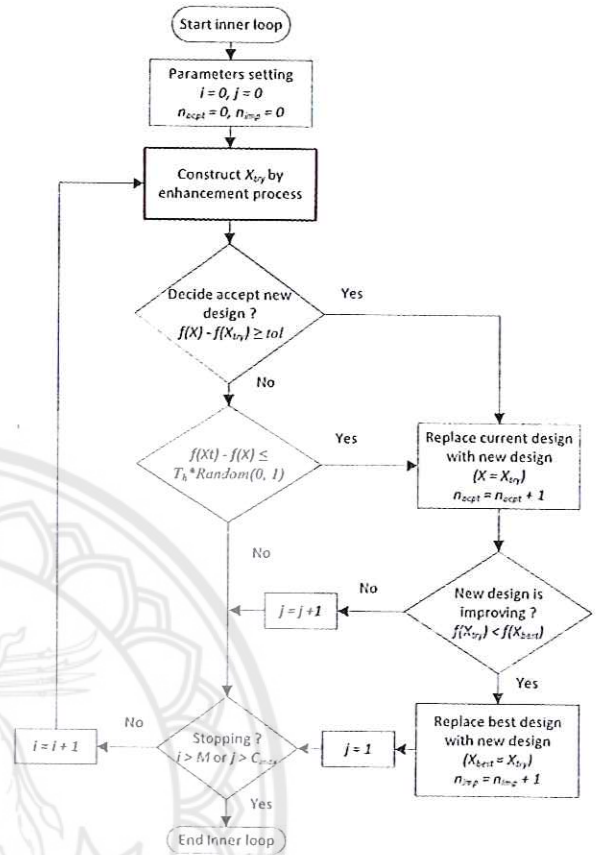


Figure 5. Flowchart of MESE inner loop

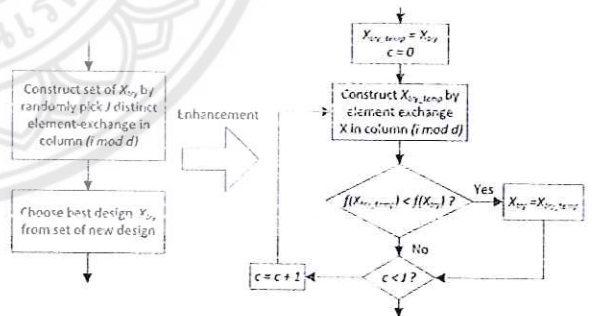


Figure 6. A new design construction using SA (Step 2)

IV. RESULTS

The values of ϕ_p criteria at the termination step of MSA, ESE and MESE from each dimension of problems are presented in Table 2. Each case study was repeated for 10 times to consider the effect of different starting points. The descriptive statistics on the ϕ_p values obtained from each search technique are displayed in columns 3-6. The results in columns 3-6 indicate that MSA, ESE and EMSE perform similarly for small dimension of problem in terms of minimization of ϕ_p criterion. Further, the standard deviation values appeared in column 6 displays a slightly larger amount of variation over 10 replications in ESE and EMSE than that of MSA. This indicates the consistency in the search process for MSA when different starting points are considered. When medium dimensions are considered, ϕ_p values from ESE and MESE are slightly lower than MSA. In addition, small amount of standard deviation is observed. For large dimensions of problem, both of ESE and MESE perform similar results in terms of minimization of ϕ_p values. Hence if the good property of design is concerned, either ESE or MESE can be used for constructing the optimal LHD.

TABLE 2. Descriptive statistics of ϕ_p values obtained from MSA, ESE and MESE

LHDs	Algorithm	$(p = 5, t = 2)$			
		Min	Max	Mean	SD
9 × 2	MSA	4.273	4.273	4.273	0
	ESE	4.273	4.344	4.287	0.029
	MESE	4.273	4.344	4.280	0.022
51 × 5	MSA	5.417	5.43	5.422	0.004
	ESE	5.415	5.431	5.422	0.006
	MESE	5.418	5.43	5.423	0.004
201 × 10	MSA	6.179	6.181	6.180	0.000
	ESE	6.170	6.174	6.172	0.001
	MESE	6.173	6.185	6.184	0.000
451 × 15	MSA	6.776	6.779	6.777	0.001
	ESE	6.760	6.762	6.761	0
	EESE	6.760	6.762	6.761	0
801 × 20	MSA	7.272	7.273	7.272	0
	ESE	7.253	7.254	7.254	0
	EESE	7.254	7.254	7.254	0

The results of the performance (efficiency) for MSA, ESE and MESE algorithms are presented in Table 3. This table presents the time elapsed and number of exchange

required for each algorithm to reach the same level of ϕ_p values. For each dimension of problem, the search algorithms are repeated for 10 times. Hence all values are presented as the average values. For small dimension case, it can be clearly seen that ESE and MESE converges much faster than MSA. The number of exchange required in the search process is also less than the MSA, while MESE requires less number of exchanges comparing to ESE. For medium and large dimensions of problem, MESE converges much faster than MSA while it performs slightly better than ESE. Further, the number of exchange obtained from MESE is the smallest value. This indicates that if time constraint is taken into account, MESE could be the better choice to use in the construction of the optimal LHD designs.

The results in columns 5-7 display time ratio for each search algorithm. It can be concluded from these ratio that MESE converges much more quickly than MSA. The maximum improvement over MSA can be observed when the dimension of problem is small. In the case of larger dimension, the improvement ratio turns to a small value. It could be concluded that the performance of these three algorithms are close to each other especially ESE and MESE algorithm.

TABLE 3. Performance of MSA, ESE and EMSE

LHDs	Algorithm	Performance (Average)		Time ratio		
		Time (sec.)	#Exchange	MSA / ESE	MSA/ MESE	ESE/ MESE
2 × 9	MSA	19.993	47140			
	ESE	3.595	5760	5.561	8.707	1.565
	MESE	2.296	5415			
51 × 5	MSA	751.54	284931			
	ESE	550.762	150000	1.364	2.400	1.759
	MESE	313.067	118950			
201 × 10	MSA	2795.741	209912			
	ESE	1349.788	17580	2.071	2.376	1.147
	MESE	1176.529	94070			
451 × 15	MSA	8686.660	234517	2.380	2.463	1.034
	ESE	3648.646	185220			
	MESE	3526.691	124550			
801 × 20	MSA	20854.01	260539	2.692	2.718	1.009
	ESE	7744.424	220480			
	MESE	7672.487	185750			

V. CONCLUSIONS

This paper presents the method to enhance the SA and ESE algorithms in the construction of the optimal LHD. The major enhancement method appears in the calculation of ϕ_p criterion and the tolerance level setting in SA. For ESE, the enhancement is applied by using the combination of SA and ESE especially in the inner loop as shown in Figure 5 and 6. As presented in the result section, MESE perform better than ESE and MSA in terms of the design property achievement and the efficiency. Hence MESE would be recommended for the construction of optimal LHD for CSE. In order to extend the conclusion, other classes of design can be developed and collaborated with MESE to search for the best design in the class. Further, other types of search algorithm like Particle swarm optimization (PSO) or Ant colony can be further developed in constructing an optimal LHD or other classes of space filling design. The validation of the approximation model accuracy developed from the obtained optimal design could also be further investigated.

REFERENCES

- [1] R. A. Bates, R. J. Buck, E. Riccomagno, and H. P. Wynn, "Experimental design and observation for large systems". *Journal of the royal statistical society, Series B*, 58, 1996, pp. 77-94.
- [2] N. A. Butler, "Optimal and orthogonal latin hypercube designs for computer experiments", *Biometrika*, 88(3), 2001, pp. 847-857.
- [3] A. Grosso, A. Jamali, and M. Locatelli, "Finding maximin latin hypercube designs by iterated local search heuristics", *European Journal of Operation Research*, vol. 197(2), 2009, pp. 541-547.
- [4] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments". *Journal of Statistical Planning and Inference*, vol. 134, 2005, pp. 268-287.
- [5] M. E. Johnson, I. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs". *Journal of Statistical Planning and Inference*, vol. 26, 1990, pp. 131-148.
- [6] J. Koehler, and A. B. Owen, "Computer experiments. *Handbook of Statistics*", vol. 13, Elsevier Science, New York, 1996, 261-308.
- [7] S. Leary, A. Bhaskar and A. Keane, "Optimal orthogonal-array-based latin hypercubes". *Journal of Applied Statistics*, vol. 30(5), 2003, pp. 585-598.
- [8] W. Li, and C.F.J. Wu. "Columnwise-pairwise algorithms with applications to the construction of supersaturated designs" *Technometrics*, vol. 39, 1997, pp. 171-179.
- [9] Z. Li and N. Shigeru, "Maximin distance-lattice hypercube design for computer experiment based on genetic algorithm," *IEEE explore*, vol. 2, 2001, pp. 814-819.
- [10] M. Liefvandahl and R. Stocki, "Study on algorithms for optimization of latin hypercubes," *Journal of Statistical Planning and Inference*, vol. 136, 2006, pp. 3231-3247.
- [11] M.D. Mackay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, 1979, pp. 239-246.
- [12] M.D. Morris, and T. J. Mitchell, "Exploratory design for computational experiments," *Journal of Statistical Planning and Inference*, vol. 43, 1995, pp. 381-402.
- [13] J. S. Park, "Optimal latin hypercube designs for computer experiments," *Journal of Statistical Planning and Inference*, vol. 39, 1994, pp. 95-111.
- [14] R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org/>.
- [15] Y. G. Saab, and Y. B. Rao, "Combinatorial optimization by stochastic evolution," *IEEE Transaction on Computer-aided Design*, vol.10, 1991, pp. 525-535, 1991.
- [16] J. Sacks, W. J. Welch, T.J. Mitchell, and H. P. Wynn, "Design and analysis of computer experiments," *Statistical Science*, vol. 4(4), 1989, pp. 409-435.
- [17] M. Shewry, and H. P. Wynn, "Maximin entropy design," *Journal of applied statistics*, vol. 14(2), 1987, pp. 165-170.
- [18] F. A. C. Viana, G. Venter, V. Balanov, "An algorithm for fast optimal latin hypercube design of experiments," *International Journal for Numerical Methods in Engineering*, vol. 82(2), 2010, pp. 135-156.
- [19] W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris, "Screening, predicting, and computer experiments," *Technometrics*, vol. 34, 1992, pp. 15-25.
- [20] K. Q. Ye, W. Li, and A. Sudjianto, "Algorithmic construction of optimal symmetric latin hypercube designs," *Journal of Statistical Planning and Inference*, vol. 90, 2000, pp. 145-159.