



**อภิธานการ**

การสร้างแผนการทดลองที่เหมาะสมโดยใช้อัลกอริทึมแก้ไข  
วิวัฒนาการการปรับปรุง

สำนักหอสมุด

Construction of the optimal designs using modified  
evolutionary search algorithm



สำนักหอสมุด มหาวิทยาลัยนครสวรรค์
วันลงทะเบียน.....2.2.7.0.2554....
เลขทะเบียน.....1588669x.....
เลขเรียกหนังสือ.....9. 9A.....

จรัสศรี รุ่งรัตนอุบล<sup>1</sup>  
อนามัย นาอุดม<sup>2</sup>

๗,๑  
๐๓๕  
๖๑๖๓  
๒๕๖๒

<sup>1</sup>ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

<sup>2</sup>ภาควิชาคณิตศาสตร์

คณะวิทยาศาสตร์

มหาวิทยาลัยนครสวรรค์

## สารบัญ

ประกาศคุณูปการ .....	II
บทคัดย่อ .....	III
Abstract .....	IV
บทที่ 1 .....	5
บทนำ .....	5
บทที่ 2 .....	15
เอกสารและงานวิจัยที่เกี่ยวข้อง .....	15
การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE) .....	15
อัลกอริทึมการสืบค้นและเกณฑ์การเลือกค่าที่เหมาะสม .....	21
บทที่ 3 .....	32
วิธีดำเนินการวิจัย .....	32
บทที่ 4 .....	44
ผลการวิจัย .....	44
บทที่ 5 .....	60
สรุปผลการวิจัย .....	60
ภาคผนวก .....	65

## ประกาศคุณประการ

งานวิจัยนี้สำเร็จลุล่วงได้ด้วยดีด้วยความกรุณาอย่างยิ่งจากคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวรที่ได้ให้ทุนสนับสนุนในการทำวิจัย

ขอขอบคุณ ดร. อนามัย นาอุดมผู้ร่วมโครงการที่ให้ความช่วยเหลือด้านสถิติวิเคราะห์ การโปรแกรม และนายธัญดล จันทร์วงศ์ ผู้ให้ความร่วมมือในการเขียนโปรแกรมและเขียนบทความวิจัย เพื่อนำเสนอในงานประชุมวิชาการ The 6th National Conference on Computing and Information Technology ที่จัดขึ้น ณ โรงแรมอีสเทิร์น กรุงเทพมหานคร ในวันที่ 3-5 มิถุนายน 2553

นอกจากนี้ข้าพเจ้าใคร่ขอขอบคุณ นิสิตช่วยวิจัยทุกท่านที่ให้ความร่วมมืออย่างดีจนทำให้งานวิจัยนี้ประสบความสำเร็จอย่างน่าพอใจยิ่ง

คุณค่าและประโยชน์อันพึงมีจากงานวิจัยนี้ ผู้วิจัยขอมอบอุทิศแด่ผู้มีพระคุณทุก ๆ ท่าน

จรัสศรี รุ่งรัตนอุบล

มิถุนายน 2553



ชื่อเรื่อง การสร้างแผนการทดลองที่เหมาะสมโดยใช้อัลกอริทึมเชิงวิวัฒนาการการปรับปรุง

ชื่อเรื่องภาษาอังกฤษ Construction of the optimal designs using modified evolutionary search algorithm

หัวหน้าโครงการวิจัย ผู้ช่วยศาสตราจารย์ ดร. จรัสศรี รุ่งรัตนอุบล

ผู้ร่วมวิจัย ผู้ช่วยศาสตราจารย์ ดร. อนามัย นาอุดม

งบประมาณวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวรประจำปี 2552

### บทคัดย่อ

การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer simulated experiment: CSE) ได้ถูกนำมาใช้อย่างแพร่หลายในการศึกษารูปแบบความสัมพันธ์ของตัวแปรเข้า (Input variables) และตัวแปรออก (Output response) ของระบบที่มีความซับซ้อน ประสิทธิภาพและความน่าเชื่อถือของผลลัพธ์ที่ได้จากการจำลองการทดลองด้วยคอมพิวเตอร์จะขึ้นอยู่กับ การวางแผนเลือกกลุ่มของตัวแปรเข้าเพื่อทำการประมวลผลผ่านกระบวนการจำลอง (Simulation routine) และได้ค่าของตัวแปรออก โดยทั่วไปการเลือกตัวแปรเข้าหรือการวางแผนการทดลอง (Design of experiments) จะใช้วิธีการค้นหาแผนการทดลองที่เหมาะสม (Optimal design) โดยใช้อัลกอริทึมการค้นหา (Search algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria) ในงานวิจัยนี้ผู้วิจัยจะประยุกต์ใช้อัลกอริทึมการค้นหาเชิงวิวัฒนาการแบบปรับปรุง (Modified evolutionary search algorithm) เพื่อสร้างแผนการทดลองที่เหมาะสมบนคลาสของแผนการทดลองแบบละตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) โดยพิจารณาเกณฑ์ในการเลือกค่าที่เหมาะสมที่นิยมใช้ในงานด้าน CSE หลายเกณฑ์ จากนั้นทำการเปรียบเทียบประสิทธิภาพของวิธีการที่นำเสนอ กับอัลกอริทึมการค้นหาที่ใช้อย่างแพร่หลาย เช่น Columnwise-pairwise algorithm (CP) และ Simulated annealing algorithm เป็นต้น โดยพิจารณาจากอัตราการลู่เข้า (Rate of convergence) และคุณสมบัติของแผนการทดลอง (Design property) ที่เหมาะสมจากกระบวนการสืบค้น

## Abstract

Currently the relation of input variables and output response is investigated through the computer simulated experiments (CSE). Such experiments are usually time-consuming and computationally expensive to run. Therefore, a selection of settings of input variables to run CSE is crucially critical. The space filling designs such as Latin hypercube designs (LHD) are normally practiced in the context of CSE. The best design for a given dimensional problem is obtained by using a search algorithm under a pre-specified optimality criterion. Normally these search processes take a long time to terminate, especially when the dimension of the problem is large and sometimes the process is terminated before the best design is achieved. The aim of this research is to propose the modified evolutionary search algorithm to construct the optimal LHD. The performance of the proposed method will be compared to the popular search algorithm like Columnwise-pairwise algorithm (CP) and Simulated annealing algorithm (SA). The comparative studies will be made based on the rate of convergence and the design property of the best searched design.

# บทที่ 1

## บทนำ

ชื่อเรื่อง	การสร้างแผนการทดลองที่เหมาะสมโดยใช้อัลกอริทึมเชิงวิวัฒนาการการปรับปรุง
ชื่อเรื่องภาษาอังกฤษ	Construction of the optimal designs using modified evolutionary search algorithm
หัวหน้าโครงการวิจัย	ผู้ช่วยศาสตราจารย์ ดร. จรัสศรี รุ่งรัตนอุบล (50%)
ผู้ร่วมวิจัย	ผู้ช่วยศาสตราจารย์ ดร. อนามัย นาอุดม (50%)

### 1. ข้อมูลทั่วไป

ประเภทโครงการ  วิจัยพื้นฐาน<sup>1</sup>  วิจัยประยุกต์<sup>2</sup>  วิจัยเชิงพัฒนา<sup>3</sup>  
เทคโนโลยีเป้าหมาย งานวิจัยนี้เกี่ยวข้องกับการพัฒนาเทคนิคการวางแผนทดลอง  
เพื่อใช้ใน การจำลองการทดลองทางวิศวกรรมศาสตร์ และ  
คณิตศาสตร์ประยุกต์ การจำลองการทดลองทางวิศวกรรมศาสตร์ และ  
อุตสาหกรรมเป้าหมาย

ระดับความสอดคล้องของโครงการวิจัยกับนโยบายและยุทธศาสตร์การวิจัยของชาติ (พ.ศ.2551-2553)

1. การสร้างศักยภาพและความสามารถในการพัฒนาทางเศรษฐกิจ

2. การสร้างศักยภาพและความสามารถในการพัฒนาทางสังคม

3. การสร้างศักยภาพและความสามารถในการพัฒนาทางวิทยาการและทรัพยากร

4. การเสริมสร้างและพัฒนาทุนทรัพยากรธรรมชาติและสิ่งแวดล้อม

5. การบริหารจัดการความรู้ ผลงานวิจัย ทรัพยากร และภูมิปัญญาของประเทศ สู่  
การใช้ประโยชน์ด้วยยุทธวิธีที่เหมาะสม

<sup>1</sup> การวิจัยพื้นฐาน [Basic Research] หมายถึง การวิจัยที่มุ่งสร้างฐานความรู้ เพื่อให้ได้ข้อมูลในการนำไปศึกษาค้นคว้าต่อไป

<sup>2</sup> การวิจัยประยุกต์ [Applied Research] หมายถึง การวิจัยที่มีจุดมุ่งหมายที่แน่นอนในการนำผลงานไปประยุกต์ใช้

<sup>3</sup> การวิจัยเชิงพัฒนา [Development Research] หมายถึง การวิจัยที่มุ่งนำความรู้หรือผลที่ได้จากการวิจัยพื้นฐานและการวิจัยประยุกต์ไปพัฒนาเชิงอุตสาหกรรมหรือเชิงพาณิชย์กรรม

2. คณะผู้วิจัย<sup>4</sup> (แสดงรายชื่อหัวหน้าโครงการ/นักวิจัยร่วม/ผู้ช่วยวิจัย)

ชื่อ ดร. จรัสศรี รุ่งรัตนอุบล

ตำแหน่ง ผู้ช่วยศาสตราจารย์                      คุณวุฒิ/สาขาวิชา                      PhD (Computer Science)

สถานที่ติดต่อ ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์

มหาวิทยาลัยนเรศวร

โทรศัพท์ (055) 963205                      โทรสาร (055) 963260

โทรศัพท์                      0840486173                      E-mail: jaratsrir@nu.ac.th

ความชำนาญ: การเขียนโปรแกรมด้วยภาษาต่าง ๆ

ประสบการณ์ที่เกี่ยวข้องกับงานในโครงการที่เสนอ: กำลังทำงานวิจัยที่เกี่ยวข้องกับอัลกอริทึมการสืบค้นแบบต่าง ๆ เพื่อนำไปใช้ในการวางแผนการทดลองเพื่อการจำลองด้วยคอมพิวเตอร์

หน้าที่ในโครงการที่เสนอ(ระบุส่วนงาน) เป็นหัวหน้าโครงการวิจัย

หน้าที่ในโครงการที่เสนอดคิดเป็น                      50% ของงานในโครงการ (คิดทั้งโครงการ = 100%)

เวลาที่ใช้ในโครงการนี้คิดเป็น                      30% ของเวลางานปกติ (โดยคิดเทียบเวลาทำการ 7 ชั่วโมง)

เวลาที่ใช้ในโครงการอื่นๆ ซึ่งกำลังเสนอขอทุน/ยังอยู่ระหว่างดำเนินการ (ถ้ามีโปรดระบุ)

ชื่อ ดร. อนามัย นาคุดม

ตำแหน่ง ผู้ช่วยศาสตราจารย์                      คุณวุฒิ/สาขาวิชา                      Ph.D. (Applied Statistics)

สถานที่ติดต่อ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

โทรศัพท์ (055) 963259                      โทรสาร (055) 963201                      E-mail: anamain@nu.ac.th

ความชำนาญ: การโปรแกรมบน R package และการสร้างตัวแบบทางสถิติแบบต่าง ๆ

ประสบการณ์ที่เกี่ยวข้องกับงานในโครงการที่เสนอ: ทำวิจัยที่เกี่ยวข้องกับการสร้างแผนการทดลองที่เหมาะสมและพัฒนาตัวแบบทางสถิติเพื่อการจำลองการทดลองด้วยคอมพิวเตอร์

หน้าที่ในโครงการที่เสนอ (ระบุส่วนงาน) เป็นนักวิจัยร่วม

หน้าที่ในโครงการที่เสนอดคิดเป็น                      50% ของงานในโครงการ (คิดทั้งโครงการ = 100%)

เวลาที่ใช้ในโครงการนี้คิดเป็น                      20% ของเวลางานปกติ (โดยคิดเทียบเวลาทำการ 7 ชั่วโมง)

3. วัตถุประสงค์

1) เพื่อเสนอวิธีการสร้างแผนการทดลองที่เหมาะสมบนคลาสของ LHD โดยการประยุกต์ใช้

อัลกอริทึมเชิง                      วิวัฒนาการ

2) เพื่อพัฒนาวิธีการสร้างแผนการทดลองที่เหมาะสมให้มีประสิทธิภาพมากยิ่งขึ้นโดยผนวกวิธีการ

ทางพีชคณิตเข้า                      กับวิธีฮิวริสติก

<sup>4</sup> โปรดส่งประวัติเต็ม(CV) ของคณะผู้วิจัยทั้งหมด จำนวน 1 ชุด (แยกจากเอกสารข้อเสนอโครงการ)

3) เพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึมการสืบค้นที่เสนอขึ้นใหม่กับอัลกอริทึมที่มีอยู่แล้วว่ามีข้อเด่น/ ข้อด้อยแตกต่างกันอย่างไร

#### 4. เป้าหมาย

งานวิจัยนี้เป็นงานวิจัยต่อเนื่องที่มีวัตถุประสงค์ในปรับปรุงประสิทธิภาพของอัลกอริทึมการสืบค้นและนำเสนอวิธีการใหม่ ๆ ในขั้นตอนการทำงานของอัลกอริทึมการสืบค้น เช่น การกำหนดแผนการทดลองตั้งต้น การกำหนดกฎการแทนที่ และการกำหนดกฎการหยุดสืบค้นแบบต่าง ๆ เพื่อให้อัลกอริทึมที่ถูกนำเสนอมีประสิทธิภาพในการค้นหาแผนการทดลองที่เหมาะสมมากยิ่งขึ้น ซึ่งจะใช้หลักการผสมผสานระหว่างวิธีการทางพีชคณิต (Algebraic method) กับวิธีฮิวริสติก (Heuristic method) เข้าด้วยกัน ซึ่งจะส่งผลให้สามารถสร้างแผนการทดลองที่เหมาะสมได้รวดเร็วยิ่งขึ้น

#### 5. ผลกระทบโดยรวมต่อประเทศ

งานวิจัยนี้จะมีผลกระทบในเชิงวิชาการและการศึกษา เนื่องจากผู้ใช้สามารถนำผลจากการวิจัยไปใช้ในการค้นหาแผนการทดลองที่เหมาะสมที่สุดเมื่อทราบมิติของปัญหาใด ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น โดยการใช้อัลกอริทึมการสืบค้นที่พัฒนาขึ้นนี้จะมีจุดเด่นคือมีอัตราการลู่เข้าเร็วขึ้น และสามารถหาแผนการทดลองที่เหมาะสมเชิงกลุ่ม (Global optimum) ได้

#### 6. หลักการและเหตุผลที่เสนอโครงการ

ในปัจจุบันการจำลองการทดลองด้วยคอมพิวเตอร์ได้เข้ามามีบทบาทในการศึกษารูปแบบความสัมพันธ์ของตัวแปรเข้า (Input variables) และตัวแปรออก (Output response) มากขึ้น โดยเฉพาะอย่างยิ่งในกรณีที่ไม่สามารถทำการทดลองทางกายภาพ (Physical experiment) ได้ อันเนื่องมาจากข้อจำกัดของวัสดุทดลอง (Experimental material) เช่น การจำลองตัวแบบ 3 มิติทางคณิตศาสตร์ผ่านตัวจำลองคอมพิวเตอร์เพื่อวิเคราะห์หาปริมาณน้ำมันใต้ดินในบางพื้นที่เมื่อทราบคุณลักษณะบางประการของตัวแปรเข้าที่เกี่ยวข้องในพื้นที่นั้น ๆ [19] หรือการใช้การคำนวณพลศาสตร์ของไหล (Fluid dynamics method) ศึกษาลักษณะการไหลเวียนของโลหิตเมื่อทำการทดลองฉีดยาให้กับผู้ป่วย เป็นต้น จะเห็นได้ว่าการทดลองลักษณะนี้ ไม่สามารถทำการทดลองกับหน่วยทดลองจริง ๆ ได้ เนื่องจากข้อจำกัดด้านค่าใช้จ่ายและความปลอดภัยของหน่วยทดลอง จึงต้องหลีกเลี่ยงการวางแผนทดลองทางกายภาพโดยการใช้ CSE เข้ามาแทนที่เพื่อช่วยในการศึกษาคุณลักษณะของระบบที่สนใจ โดยทั่วไปแล้วเทคนิคการจำลองด้วยคอมพิวเตอร์หรือ CSE มักจะใช้เวลาและสิ้นเปลืองทรัพยากรมาก [16, 19, 23] ดังนั้นนักวิจัยทั่วไปจึงได้มีการคิดค้นที่จะค้นหากลุ่มของระดับตัวแปรเข้า (Treatment combination) หรือแผนการทดลองที่เหมาะสมไปใช้ในการจำลองการทดลองด้วยคอมพิวเตอร์เพื่อให้แน่ใจได้ว่าจุดทดลองต่าง ๆ ครอบคลุมปริภูมิของการทดลอง (Design space) ทั้งหมดที่สนใจศึกษา ในการสร้างแผนการทดลองที่เหมาะสม (Optimal design) จะต้องใช้อัลกอริทึมการสืบค้น (Search algorithm) [5, 8, 12, 13, 16] ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria) [5, 9]



ลักษณะเฉพาะที่สำคัญอย่างหนึ่งของ CSE คือ ตัวแปรออกซึ่งเป็นผลลัพธ์ที่ได้จากการประมวลการจำลองจะมีลักษณะแบบ Deterministic ดังนั้นแผนการทดลองที่เหมาะสมกับ CSE จะต้องเป็นแผนการทดลองที่มีลักษณะเติมเต็มปริภูมิ (Space filling) ซึ่งแผนการทดลองที่มีคุณสมบัติดังกล่าวถูกนำมาใช้อย่างแพร่หลายในงานด้าน CSE คือ แผนการทดลองแบบละตินไฮเปอร์คิวบ์ (Latin hypercube design: LHD) ซึ่งถูกนำเสนอขึ้นในปี ค.ศ. 1979 โดย McKay et al. [14] ดังนั้นงานวิจัยนี้ ผู้วิจัยจึงมีความสนใจศึกษาหลักการสร้างแผนการทดลองที่เหมาะสม (Optimal design) สำหรับ CSE โดยเน้นศึกษาบนคลาสของ LHD

แผนการทดลองแบบ LHD คือเมตริกซ์  $X$  ซึ่งประกอบไปด้วย  $n$  แถว และ  $d$  คอลัมน์ เมื่อ  $n$  คือจำนวนจุดทดลอง และ  $d$  คือจำนวนตัวแปรเข้าในแผนการทดลอง ซึ่งในทางปฏิบัตินั้น การสร้างแผนการทดลองแบบ LHD สามารถทำได้โดยการเรียงสับเปลี่ยนแบบสุ่ม (Random permutation) ของตัวเลข 1 ถึง  $n$  ในแต่ละคอลัมน์ จากนั้นนำตัวเลขที่ได้จากทั้งหมด  $d$  คอลัมน์มาผนวกเข้าด้วยกัน เป็นเมตริกซ์ของการทดลอง  $X$  ซึ่งวิธีการนี้จะทำให้การกระจายของแต่ละจุดทดลองจะเป็นไปโดยสุ่ม และการจัดการเพื่อให้จุดทดลองกระจายทั่วปริภูมิของการทดลอง (Design space) สามารถทำได้โดยใช้เทคนิคของอัลกอริทึมการสืบค้น (Search algorithm) [3, 5, 8, 10, 12, 16] ยกตัวอย่างเช่น Columnwise-pairwise algorithm (CP) [12], Simulated annealing algorithm (SA) [16] และ Genetic algorithm (GA) [13] ควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria) [5, 9, 19] เช่น Integrated Mean Square Error (IMSE) เกณฑ์เอนโทรปี (Entropy criterion) หรือ เกณฑ์  $\phi_p$  เป็นต้น

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 ดังนั้นปัญหาการหาค่าเหมาะสมในการสร้างแผนการทดลองแบบเติมเต็มปริภูมิ คือการหาแผนการทดลอง  $x^*$  ใด ๆ ที่สอดคล้องกับเงื่อนไขต่อไปนี้

$$h(x^*) = \min_{X \in R^d} h(X) \quad (2)$$

ทั้งนี้ฟังก์ชันวัตถุประสงค์  $h(\cdot)$  อาจหมายถึงเกณฑ์เลือกค่าเหมาะสมใด ๆ ซึ่งจะต้องมีการกำหนดไว้ล่วงหน้าก่อนจะทำการค้นหา

ตามที่ได้กล่าวไว้ข้างต้นแล้วว่า อัลกอริทึมการสืบค้นที่ถูกนำมาใช้ในงานด้าน CSE อย่างแพร่หลาย ประกอบไปด้วย CP, SA และ GA ซึ่งอัลกอริทึมเหล่านี้มีขั้นตอนการทำงานที่คล้ายคลึงกัน และสามารถสรุปหลักการทำงานของอัลกอริทึมเหล่านี้ ได้ดังต่อไปนี้

ขั้นตอนที่ 1: เริ่มต้นการค้นหาโดยการสุ่มเลือกแผนการทดลองตั้งต้น  $x^0$  และกำหนดให้  $x^c = x^0$

ขั้นตอนที่ 2: สร้างโครงสร้างย่านใกล้เคียง (Neighborhood structure) ของ  $x^c$  ตามนิยามของโครงสร้างที่ไว้ล่วงหน้า 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 ซึ่งจะถือว่าเป็นแผนการทดลองที่เหมาะสมจากกระบวนการสืบค้น

จากขั้นตอนการทำงานของอัลกอริทึมหาค่าเหมาะสมที่กล่าวมาข้างต้น รวมไปถึงข้อจำกัดด้านเวลา ในกรณีที่ต้องการวางแผนการทดลองที่เหมาะสม สำหรับปัญหาที่มีมิติขนาดใหญ่ขึ้น ผู้วิจัยจึงสนใจศึกษา หาวิธีการปรับปรุงประสิทธิภาพของอัลกอริทึมเหล่านี้ เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้น การกระจายจุดทดลองให้ทั่วปริภูมิของแผนทดลอง นักวิจัยได้ประยุกต์ใช้อัลกอริทึมการสืบค้น (Search algorithm) ควบคู่กับเกณฑ์ในการเลือกค่าเหมาะสม (Optimality criteria) เพื่อสร้างแผนการทดลองที่เหมาะสมและมีคุณสมบัติความเต็มเต็มปริภูมิ (Space filling) และเชิงตั้งฉาก (Orthogonality) ดังที่ได้รายงานไว้ในงานวิจัยหลายชิ้น [3, 5, 6, 8, 10, 11, 20, 22] ซึ่งรายงานไว้ว่าแผนการทดลองแบบละตินไฮเปอร์คิวบ์ที่เหมาะสม (Optimal Latin hypercube design: OLHD) เป็นแผนการทดลองที่มีประสิทธิภาพดีกว่าแผนการทดลองแบบ Simple random sequence ทั่ว ๆ ไป แต่ทั้งนี้การค้นหาแผนการทดลอง LHD ที่เหมาะสม หรือ OLHD นั้นไม่ใช่เรื่องง่าย โดยเฉพาะอย่างยิ่งในกรณีที่มีมิติของแผนการทดลองมีขนาดใหญ่ขึ้น [5, 8, 11, 16, 18] การค้นหาแผนการทดลองที่เหมาะสมโดยใช้อัลกอริทึมการสืบค้นควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสมนั้นได้ถูกนำเสนอไว้หลายวิธี เช่น Morris and Mitchell [16] ได้พัฒนาอัลกอริทึม Simulated annealing (SA) ภายใต้เกณฑ์  $\phi_p$  จากนั้น Li and Wu [12] ได้เสนอการค้นหาแผนการทดลองที่เหมาะสมโดยใช้อัลกอริทึม CP ภายใต้เกณฑ์ Integrated mean square error (IMSE) และ เกณฑ์เอนโทรปี (Entropy criteria) ต่อมา Jin et al. [8] ได้ดัดแปลงอัลกอริทึมวิวัฒนาการแบบสุ่ม (Enhanced stochastic evolutionary algorithm: ESE) เพื่อทำการค้นหาแผนการทดลองที่เหมาะสมโดยพิจารณาเกณฑ์ที่หลากหลายเช่น เกณฑ์ Maximin distance เกณฑ์  $\phi_p$  และเกณฑ์ Entropy ในปีถัดมา Liefvendahl and Stocki [13] ประยุกต์อัลกอริทึมแบบเจเนติก (Genetic algorithm: GA) เพื่อทำการค้นหาแผนการทดลองที่ดีที่สุด ภายใต้เกณฑ์  $\phi_p$  และ เกณฑ์ Maximin จากงานวิจัยของ Rungrattanaubol and Na-udom [18] ได้ทำการเปรียบเทียบประสิทธิภาพของ SA และ GA ในแง่ของความเร็วในการลู่เข้าสู่ค่าที่เหมาะสม (Rate of convergence) และพบว่า SA สามารถทำงานได้ดีกว่า GA เกือบทุกกรณีของการศึกษา นอกจากนี้ Joseph and Hung [10] เสนอการสร้างแผนการทดลองแบบ LHD โดยพิจารณาเกณฑ์ Maximin distance และเกณฑ์ Minimum pair-wise correlation ควบคู่กันโดยใช้อัลกอริทึมที่พัฒนาขึ้นมาใหม่และเปรียบเทียบผลที่ได้กับงานวิจัยที่ได้รายงานไว้ใน Morris and Mitchell [16] พบว่าประสิทธิภาพของแผนการทดลองในเชิง

ของคุณสมบัติที่พิจารณาพอ ๆ กัน ส่วนงานวิจัยของ Cioppa and Lucas [3] ได้เสนออัลกอริทึมสำหรับสร้างแผนการทดลอง OLHD ที่มีมิติใหญ่ขึ้นเมื่อเทียบกับงานวิจัยก่อนหน้า นอกจากนี้ยังพบว่าวิธีที่นำเสนอขึ้นมาก่อให้เกิดแผนการทดลองที่มีคุณสมบัติที่ดีมากในแง่ของความเต็มเต็มปริภูมิและคุณสมบัติเชิงตั้งฉาก จากนั้น Steinberg [20] เสนอการวางแผนการทดลองแบบ Orthogonal LHD โดยพยายามทำให้ค่าสหสัมพันธ์ระหว่างคอลัมน์ของแผนการทดลองมีค่าน้อยที่สุด ซึ่งเป็นการขยายงานที่นำเสนอไว้ใน Ye [22] เพื่อให้สามารถสร้างแผนการทดลองที่มีมิติใหญ่ขึ้นได้ ส่วน Prescott [17] เสนอการสร้างแผนการทดลองแบบ LHD ที่มีค่า pair-wise correlation เท่ากับ 0 และพิจารณาว่าคุณสมบัติดังกล่าวมีความสัมพันธ์กับคุณสมบัติความเต็มเต็มปริภูมิหรือไม่ อย่างไรก็ตามข้อจำกัดของงานวิจัยชิ้นนี้คือวิธีการที่นำเสนอสามารถทำได้เฉพาะกรณีมิติของปัญหาขนาดเล็กเท่านั้น ในส่วนของการเสนออัลกอริทึมเพื่อสร้างแผนการทดลอง OLHD สามารถดูได้จาก Grosso et al. [6] เสนอวิธีสร้างแผนการทดลองแบบ LHD ที่เหมาะสมโดยใช้ Iterated Local Search (ILS) ซึ่งผลที่ได้พบว่าวิธีการดังกล่าวสามารถทำได้ดีเมื่อเปรียบเทียบกับงานที่เสนอไว้ใน Morris and Mitchell [16] และงานของ Jin et al. [8]

ในงานวิจัยนี้ผู้วิจัยมีแนวคิดในการปรับปรุงอัลกอริทึม ESE ที่ถูกนำเสนอไว้ใน [8] ดังกล่าวให้มีความยืดหยุ่นมากขึ้นและสามารถนำไปประยุกต์ใช้กับกระบวนการสืบค้นแบบอื่น ๆ ได้ โดยผู้วิจัยจะประยุกต์วิธีการทางพีชคณิต (Algebraic method) เข้ามาช่วยในการนิยามแผนการทดลองตั้งต้น (Initial design) ซึ่งวิธีการดังกล่าวสามารถเพิ่มประสิทธิภาพในการสืบค้นของอัลกอริทึมในแง่ของการลดปริภูมิในการค้นหาหรือการลดจำนวนสมาชิกในโครงสร้างย่านใกล้เคียงนั่นเอง นอกจากนี้จะมีการปรับหลักในการคำนวณค่าเกณฑ์การเลือกค่าที่เหมาะสม และการปรับระยะทางเริ่มต้น (Threshold distance) ในการตัดสินใจว่าจะมีการแทนที่  $x^c$  ด้วย  $x^{new}$  ระหว่างการค้นหาหรือไม่ กรณีที่  $x^{new}$  มีคุณสมบัติที่ดีขึ้นหรือแยกลงเพียงเล็กน้อย ทั้งนี้เกณฑ์การเลือกค่าที่เหมาะสมที่ถูกนำมาใช้ในงานวิจัยนี้คือเกณฑ์ Maximin และเกณฑ์  $\phi_p$  เนื่องจากเป็นเกณฑ์ที่นิยมใช้อย่างแพร่หลายในงานด้าน CSE ซึ่งการประเมินผลที่ได้จากงานวิจัยนี้จะดำเนินการตามมิติปัญหาที่แตกต่างกัน และใช้ปัญหาทดสอบตามที่ได้กำหนดไว้ใน [7] จากนั้นจะทำการยืนยันผลที่ได้โดยใช้วิธีการอนุมานเชิงสถิติในการสรุปผล และเพื่อให้ผลสรุปสมบูรณ์มากยิ่งขึ้นผู้วิจัยจะทำการเปรียบเทียบผลกับแผนการทดลองทางกายภาพ เช่น แผนการทดลอง Central composite design (CCD) แผนการทดลอง Fractional factorial design (FFD) [15] และแผนการทดลองเชิงตั้งฉากที่รายงานไว้ใน Ye [23] โดยพิจารณาจากความแม่นยำในการพยากรณ์ของตัวแบบทางสถิติที่พัฒนาขึ้นจากแผนการทดลองต่าง ๆ ดังกล่าวข้างต้น

## **7. ความพร้อมด้านอุปกรณ์/เครื่องมือ และสถานที่ดำเนินงานวิจัย**

เครื่องมือที่ใช้ในงานวิจัยประกอบด้วย เครื่องคอมพิวเตอร์ประสิทธิภาพสูง ของภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

8. แผนงานวิจัย (ส่วนนี้เป็นหัวใจของข้อเสนอโครงการ ควรเขียนให้ละเอียดและชัดเจน พร้อมทั้งอ้างอิงถึงเอกสารที่ จำเป็น เนื่องจากในการประเมินโครงการจำเป็นต้องเข้าใจรายละเอียดของแผนงาน/เป้าหมาย/ผลที่คาดว่าจะได้รับในแต่ละช่วงของกิจกรรมอย่างเพียงพอ)

ขั้นตอนแผนงาน โดยบรรยายกิจกรรมที่จะดำเนินการและระบุเป้าหมายของแต่ละขั้นตอนอย่างชัดเจน

- 1) ศึกษาลักษณะการทำงานของของ ESE อย่างละเอียดและดำเนินการกำหนดการทดลองตั้งต้นด้วยวิธีที่แตกต่างกันเช่นการใช้กฎการตั้งฉากระหว่างคอลัมน์ หรือการสมมาตรระหว่างจุดทดลอง
- 2) ศึกษาทฤษฎีและกฎการแทนที่ เช่น Threshold acceptance และกฎ SA เป็นต้น
- 3) กำหนดมิติของปัญหาที่สนใจการศึกษา
- 4) เขียนโปรแกรม R เพื่อจำลองกรณีศึกษาต่าง ๆ
- 5) ประมวลผลโปรแกรมเพื่อเก็บรวบรวมข้อมูล
- 6) วิเคราะห์ข้อมูล
- 7) สรุปประสิทธิภาพของอัลกอริทึมที่เสนอขึ้นมาใหม่กับอัลกอริทึมที่มีอยู่แล้ว เช่น CP และ SA
- 8) จัดบทความวิจัยและตีพิมพ์เผยแพร่ และจัดทำรูปเล่มรายงานวิจัยฉบับสมบูรณ์

เดือน	ผลงานที่คาดว่าจะสำเร็จ
1-3	ทราบถึงกระบวนการทำงานของอัลกอริทึมการสืบค้นแบบต่าง ๆ และทราบถึงพารามิเตอร์ที่มีความสำคัญต่อกระบวนการสืบค้น
4-6	ได้โปรแกรม R ที่สามารถจำลองข้อมูลสำหรับทุกมิติของปัญหาตามที่กำหนดไว้ในขอบเขตของการศึกษา
7-9	ได้ผลลัพธ์จากการจำลองข้อมูลกรณีต่าง ๆ และผลจากวิเคราะห์ทางสถิติ นำไปสู่ข้อสรุปสำหรับกรณีต่าง ๆ
10-12	ได้รูปเล่มรายงานที่สมบูรณ์และบทความวิจัยเพื่อทำการเผยแพร่ต่อไป

9. ผลลัพธ์ที่จะได้รับเมื่อเสร็จสิ้นโครงการ (Output): โครงการต้องระบุผลงานที่จะเกิดขึ้นจากงานวิจัยอย่างชัดเจนและเป็นรูปธรรม โดยเรียงลำดับความสำคัญของผลงานและกำหนดตัวชี้วัดเป็นจำนวนนับ (เป็นหัวข้อสำคัญในการประเมินโครงการตามเป้าหมายการผลิตผลงาน)

ผลลัพธ์	ทางตรง (หน่วยนับ)	ทางอ้อม (หน่วยนับ)
1. องค์ความรู้	เรื่อง	
2. กระบวนการใหม่	เรื่อง	
3. เทคโนโลยีใหม่	เรื่อง	
4. ดัชนีแบบ	ดัชนีแบบ	
• พร้อมใช้ (ผลิตภัณฑ์)		
• ต้องวิจัยและพัฒนาต่อ		
• ระดับห้องปฏิบัติการ		
5. สูเชิงพาณิชย์	รายการ	
6. สูเชิงพาณิชย์	รายการ	
7. สูสาธารณประโยชน์	รายการ	
8. บทความทางวิชาการ	เรื่อง	
• ผลงานตีพิมพ์ในวารสารระดับนานาชาติ		
• ผลงานตีพิมพ์ในวารสารระดับภูมิภาค		
• ผลงานตีพิมพ์ในวารสารในประเทศ		
• เอกสารประกอบการประชุม/สัมมนาในระดับนานาชาติ		
• เอกสารประกอบการประชุม/สัมมนาในระดับภูมิภาค		
• เอกสารประกอบการประชุม/สัมมนาในประเทศ	1	
• บทความในหนังสือพิมพ์และเอกสารอื่นๆ (เช่น จดหมายข่าว สื่อวิทยุ สื่อโทรทัศน์ การแถลงข่าว Media Forum นิทรรศการ/การแสดงผลงาน)		
9. สิทธิบัตร/อนุสิทธิบัตร/ลิขสิทธิ์/เครื่องหมายการค้า		

#### 10. เอกสารแนบท้าย

1. 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.

2. Castillo, E., 1988. *Extreme Value Theory in Engineering*. Academic Press, INC, San Diego.
3. Cioppa, T.M., Lucas, T.W. 2007. *Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes*. *Technometrics*, 49(1): 45-55
4. Cressie, N. A. C. 1991. *Statistics for Spatial Data*. John Wiley, New York.
5. Fang, K. T., Li, R., Sudjianto, A. 2005. *Design and modeling for computer experiments*. Chapman & Hall/CRC, London UK.
6. 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.
7. Hock, W., Schittkowski, K., 1981. *Test examples for nonlinear programming codes*. Berlin, Heidelberg, New York: Springer.
8. Jin, R., Chen, W., Sudjianto, A., 2005. *An efficient algorithm for constructing optimal design for computer experiments*. *Journal of Statistical planning and inference* 134, 268-287.
9. Johnson, M. E., Moore, I.M., and Ylvisaker, D. 1990. *Minimax and maximin distance designs*. *Journal of Statistical Planning and Inference*, 26:131-148.
10. Joseph, V.R., Hung, Y., 2008. *Orthogonal-Maximin Latin Hypercube Designs*. *Statistica Sinica*, 18: 171-186.
11. Leary, S., Bhaskar, A., and Keane, A. 2003. *Optimal orthogonal-array-based latin hypercubes*. *Journal of Applied Statistics*, 30(5): 585-598.
12. Li, W. and Wu, C.F.J. (1997). *Columnwise-pairwise algorithms with applications to the construction of supersaturated designs*. *Technometrics*, 39:171-179.
13. Liefvandahl, M. and Stocki, R. (2006). *Study on algorithms for optimization of latin hypercubes*. *Journal of Statistical Planning and Inference*, 136:3231-3247.
14. 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.
15. Montgomery, D.C., 2009. *Design and analysis of Experiments*. John Wiley & Sons, New York.
16. Morris, M.D., Mitchell, T.J., 1995. *Exploratory design for computer experiments*. *Journal of Statistical planning and inference* 43, 381-402.

17. Prescott, P., 2009. *Orthogonal-column Latin hypercube design with small samples*. Computational Statistics and Data Analysis, 53: 1191-1200.
18. 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.
19. 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.
20. Steinberg, D.M., 2006. *A construction method for orthogonal Latin hypercube designs*. Biometrika, 93(2): 279-288.
21. 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.
22. Ye, K. Q., 1998. *Orthogonal column latin hypercubes and their application in computer experiments*. Journal of the American Statistical Association 93, 1430-1439.
23. 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.

## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

การศึกษาการปรับปรุงอัลกอริทึมการสืบค้นเพื่อการจำลองการทดลองด้วยคอมพิวเตอร์ (Modification on Search Algorithm for Computer Simulated Experiment) ได้ทำการทบทวนวรรณกรรมที่เกี่ยวข้อง โดยมีเนื้อหาประกอบด้วย

การจำลองการทดลองด้วยคอมพิวเตอร์  
อัลกอริทึมการสืบค้นและเกณฑ์การเลือกค่าที่เหมาะสม  
สถิติที่เกี่ยวข้อง  
งานวิจัยที่เกี่ยวข้อง

#### การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE)

การทดลองโดยทั่วไปสามารถแบ่งเป็น 2 ประเภทใหญ่ คือ การทดลองทางกายภาพหรือการทดลองในห้องปฏิบัติการ (Physical Experiment) และ การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment) โดยส่วนมากการทดลองในชีวิตประจำวันมักจะเป็นการทดลองแบบที่ทำการทดลองกันในห้องทดลอง โรงงาน หรือแปลงเกษตร เป็นต้น ซึ่งล้วนแต่เป็นการทดลองทางกายภาพ (Physical Experiment) ซึ่งมักจะเกิดความผิดพลาดที่เกิดจากการทดลอง เนื่องจากปัจจัยด้านสภาพแวดล้อม ดังนั้นการทดลองในแต่ละครั้งอาจจะได้ผลลัพธ์ที่แตกต่างกันหรือเหมือนกัน ทั้งที่ให้ค่าตัวแปรเข้าชุดเดียวกัน (Non-Deterministic) การมีข้อผิดพลาดจากการทดลอง นั้นทำให้เกิดความซับซ้อนในการวิเคราะห์ข้อมูลและการสร้างรูปแบบจำลอง ดังนั้นผู้วิจัยต้องเลือกปัจจัยที่เกี่ยวข้องในการทดลองจำนวนน้อยๆ เพื่อง่ายในการสำรวจตรวจสอบความสัมพันธ์ระหว่างตัวแปรเข้าและผลลัพธ์ หรืออีกทางเลือกหนึ่งผู้ทำการทดลองต้องใช้ทางสถิติเข้ามาช่วยในการวางแผนการทดลอง การใช้สถิติในการวางแผนการทดลองจะอยู่บนพื้นฐานของตัวแบบทางสถิติ (Statistical Model) แผนการทดลองที่ดี จะส่งผลต่อประสิทธิภาพของตัวแบบทางสถิติที่สร้างขึ้น การวางแผนการทดลองสำหรับการทดลองทางกายภาพมีหลายแบบ เช่น Fractional Factorial Design บน ANOVA Model, Optimum Design บน Regression Model เป็นต้น ซึ่งเป็นตัวแบบที่นิยมใช้กันมาก การทดลองทางกายภาพ จะมีค่าใช้จ่ายสูงและเปลืองเวลา เพราะการทดลองทางกายภาพทำได้ยากหรือสิ้นเปลืองทรัพยากรในการทดลอง เพราะต้องคำนึงถึงปัจจัยที่เกี่ยวข้องหลากหลายปัจจัย (Fang, Li and Sudjianto, 2005)



ปัจจุบันคงปฏิเสธไม่ได้ว่า "คอมพิวเตอร์" ได้เข้ามามีบทบาทสำคัญยิ่งต่อการทำงานไม่ว่าจะเป็นสาขาอาชีพใด จะเห็นได้ว่าคอมพิวเตอร์หรือสมองกลอัจฉริยะนี้ได้เข้ามาพัฒนาและช่วยเพิ่มประสิทธิภาพการทำงานของมนุษย์ได้มากมายมหาศาล คุณสมบัติพิเศษอันโดดเด่นไม่ว่าจะเป็นประสิทธิภาพในการจัดเก็บข้อมูล ความรวดเร็วในการวิเคราะห์หรือประมวลผลและความสามารถในการทำงานซ้ำๆ กันอย่างไม่เหน็ดเหนื่อย จึงมีการนำคอมพิวเตอร์มาประยุกต์ใช้ให้เกิดประโยชน์และเหมาะสมในแต่ละงานอย่างไม่หยุดยั้ง และสิ่งเหล่านี้คือที่มาของการคิดค้นการจำลองการทำงานด้วยโปรแกรมคอมพิวเตอร์เพื่อมาใช้กับงานทางด้านวิทยาศาสตร์ และเทคโนโลยี โดยเฉพาะอย่างยิ่งใช้ในการจำลองการทดลองที่ทำได้ยากหรือทำไม่ได้จริงทางกายภาพ เช่น โปรแกรมการพยากรณ์อากาศ โปรแกรมการศึกษาพฤติกรรมของโครงสร้าง (ตึกและชิ้นส่วนอุตสาหกรรม) ภายใต้ภาวะกดดันและเงื่อนไขต่างๆ เครื่องจำลองปฏิกิริยาสำหรับวิศวกรรมปิโตรเคมีรวมไปถึงการทำงานของนิวเคลียร์ เป็นต้น เพราะข้อจำกัดต่างๆ ไม่เอื้ออำนวยให้ทำการทดลองทางกายภาพได้จริง เช่น มีงบประมาณจำกัด มีผลกระทบต่อสิ่งแวดล้อม ข้อจำกัดทางด้านกฎหมาย เทคโนโลยี ณ ปัจจุบันที่ยังไม่สนับสนุนการทดลองจริง เป็นต้น ทำให้เกิดการศึกษาทดลอง ที่เรียกว่า การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE) ขึ้นมา ซึ่งการจำลองการทดลองด้วยคอมพิวเตอร์พัฒนาก้าวหน้าเรื่อยๆ ตามการพัฒนาทางเทคโนโลยีคอมพิวเตอร์ฮาร์ดแวร์และซอฟต์แวร์ที่เจริญก้าวหน้าขึ้น

ประโยชน์ของ CSE มีด้วยกันมากมายหลายประการ ดังตัวอย่างต่อไปนี้

1. ลดการสูญเสียทรัพยากร เช่น การสร้างชิ้นส่วนปีกเครื่องบิน การซ่อมรถ การทดลองยา ถ้าใช้การจำลองเหตุการณ์จะไม่ต้องเสียค่าวัสดุอุปกรณ์ ไม่ต้องเสียทหาร ไม่ต้องเสียชีวิตคนไข้

2. ประหยัดเวลา ประหยัดทรัพย์ เพราะการทดลองในห้องทดลอง บนโต๊ะทำงาน หรือทดลองด้วยคอมพิวเตอร์จึงทำให้ค่าใช้จ่ายน้อย การแก้สมการต่างๆ ในบางครั้งการทดลองแทนค่าบนคอมพิวเตอร์จะเร็วกว่าการแก้สมการตามปกติ

3. หากสามารถสร้างแบบจำลองที่ดีได้ด้วยคอมพิวเตอร์ เราก็จะเห็นภาพหรือตัวเลขการเปลี่ยนแปลงของตัวแปรต่างๆ ในระบบเป็นขั้นเป็นตอนได้ดีกว่าการหาค่าตอบสุดท้ายแบบธรรมดา

ในงานวิจัยนี้ การจำลองการทดลองด้วยคอมพิวเตอร์ หมายถึง การใช้โปรแกรมคอมพิวเตอร์เพื่อจำลองรูปแบบของระบบใดๆ โดยอาศัยข้อมูลต่างๆ ที่เกี่ยวข้อง (Input Variable) ร่วมกับสมการทางคณิตศาสตร์ที่อธิบายการทำงานของระบบโดยได้ค่าของตัวแปรตาม (Output Variable) เพื่อใช้เป็นตัวแทนของการทำงานของระบบที่สนใจศึกษา

จากนิยามและคณะ ได้อธิบายกระบวนการในการจำลองการทดลองด้วยคอมพิวเตอร์ ว่ามีส่วนประกอบด้วยกันทั้งหมด 3 ส่วน ดังแสดงไว้ในภาพคือ

1. Experimental Design คือ การออกแบบและการวางแผนการทดลอง ซึ่งประกอบด้วยค่าของปัจจัยที่เกี่ยวข้อง (ตัวแปรเข้า) ระดับต่างๆ กล่าวคือ การออกแบบกลุ่มของค่าของตัวแปรเข้าที่ต้องการศึกษาและตามข้อจำกัดของการทดลอง (จำนวนวัน)

2. Simulation Routine หรือ กล่องดำ (Black Box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดเดาว่าเกิดอะไรขึ้นภายใน ซึ่งกระบวนการในกล่องดำ นี้จะใช้รูปแบบสมการทางคณิตศาสตร์ที่มีความซับซ้อน เพื่อให้ได้ผลลัพธ์ออกมา ซึ่งผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ

ของปัจจัยที่เกี่ยวข้องที่ป้อนเข้าไป โดยส่วนใหญ่การส่งจุดทดลองในแต่ละวัน เข้าไปในกล่องดำนั้นจะใช้  
งบประมาณและเวลานานมาก ทำให้ต้องมีการวางแผนการทดลองก่อนส่งไปประมวลผลที่กล่องดำ  
เพื่อให้ได้กลุ่มของตัวแปรตามที่มีประสิทธิภาพ

3. Approximation Model คือ การสร้างโมเดลเพื่อการประมาณเกิดจากการนำแผนการทดลอง  
กับผลลัพธ์ที่ได้หลังจากกระบวนการในกล่องดำ มาสร้างรูปแบบความสัมพันธ์ที่เหมาะสม ให้สามารถ  
นำไปใช้ประโยชน์ต่อไป จุดมุ่งหมายของการสร้างโมเดลเพื่อการประมาณ คือ การให้ได้มาซึ่งโมเดลที่  
สามารถพยากรณ์ผลลัพธ์หรือตัวแปรตามได้โดยที่ไม่จำเป็นจะต้องส่งจุดทดลองไปประมวลผลบนกล่อง  
ดำ ทำให้ประหยัดงบประมาณและเวลาในการทำงาน (Simpson, *et al.* , 2001)

### การวางแผนการทดลอง (Experimental Design)

การวางแผนการทดลองมีความสำคัญอย่างมาก ทั้งนี้เพราะแผนการทดลองที่ดีย่อมนำมาซึ่ง  
ผลลัพธ์หรือโมเดลที่มีความน่าเชื่อถือสูง เพื่อสามารถใช้เป็นตัวแทนของระบบจริงได้ ซึ่งการวางแผน  
การทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ จะแตกต่างจากการวางแผนการทดลอง  
ทางกายภาพ เพราะการจำลองการทดลองด้วยคอมพิวเตอร์จะเป็นรูปแบบที่แน่นอน (Deterministic)  
คือ ที่แต่ละครั้งใส่ตัวแปรเข้าใดๆ มีค่าเหมือนกันไปก็จะได้ผลลัพธ์หรือตัวแปรตามค่าเดิมทุกครั้ง  
(Simpson, *et al.*, 2001, Rungrattanaubol and Na-udom, 2007) การจำลองการทดลองด้วย  
คอมพิวเตอร์จะไม่มีปัจจัยภายนอกที่จะทำให้ค่าเปลี่ยนแปลงเหมือนการทดลองทางกายภาพ ทำให้การ  
วางแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ก็จะแตกต่างกับการวางแผนการ  
ทดลองทางกายภาพ แชคและคณะได้เสนอ แนวคิดของการวางแผนการทดลอง สำหรับ CSE ว่าจะต้อง  
มีความแตกต่างจากแผนการทดลองแบบคลาสสิก เนื่องจากผลลัพธ์จาก CSE เป็นแบบแน่นอน จึงไม่  
จำเป็นต้องมีการทำซ้ำ (Replicates) ดังนั้นการออกแบบการทดลองด้วยคอมพิวเตอร์จะเป็นการ  
ออกแบบโดยใช้แนวคิดแบบใหม่ ซึ่งเป็นแบบกระจายจุดทดลองให้ทั่ว Design Space ให้ได้มากที่สุด ซึ่ง  
เรียกว่า Space Filling Design ซึ่งเหมาะกับรูปแบบการทำงานแบบแน่นอน (Deterministic) (Sacks, *et al.*, 1989) การวางแผนการทดลองในลักษณะนี้เป็นแบบ Search Based ซึ่งใช้อัลกอริทึมการค้นหา  
มาเลือกแผนการทดลองที่ดีที่สุดที่กระจายจุดทดลอง ครอบคลุม Design Space ทั้งหมด ภายใต้กับ  
เกณฑ์หาค่าเหมาะสม (Optimality Criteria)

ดังได้กล่าวไปแล้วในการวางแผนการทดลองสำหรับ CSE นั้น สิ่งหนึ่งที่ผู้ออกแบบต้องคำนึง  
เสมอ คือ ลักษณะการแบ่งช่วงของค่าข้อมูลของแต่ละตัวแปรเข้า ซึ่งในการวางแผนนั้น ได้มีผู้วิจัยหลาย  
ท่านได้นำเสนอรูปแบบหรือคลาสของการวางแผนไว้หลายค่า

### คลาสของการวางแผน (Class of Design)

รูปแบบของการวางแผนการทดลองสำหรับ Space Filling ของการจำลองการทดลองด้วย  
คอมพิวเตอร์นั้นมีหลายแบบ โดยในงานวิจัยนี้จะยกตัวอย่างมาทั้งหมด 3 แบบคือ Latin Hypercubes

Design (LHD) Hammersley Sequence Sampling (HSS) และ Uniform Design ซึ่งเป็นแบบที่นิยมใช้ในการวางแผนการทดลองในปัจจุบัน

### 1. Latin Hypercubes Design (LHD)

ใช้ครั้งแรกโดย McKay, et al. ใน 1979 เป็นเมทริกซ์ ( $X$ ) มีมิติ  $n \times d$  โดยที่  $n$  คือ จำนวนรัน และ  $d$  คือ จำนวนตัวแปรเข้า ซึ่งแต่ละหลัก  $d$  ประกอบด้วยรันตั้งแต่  $\{1, 2, \dots, n\}$  เรียงสับเปลี่ยนกัน และสามารถเทียบกับค่าจริงของแต่ละตัวแปรได้ เมื่อใช้การแบ่งช่วงของแต่ละตัวแปรในพื้นที่การออกแบบ (Design Space) แล้ว จะได้เป็นค่าที่ไม่ซ้ำกันในแต่ละคอลัมน์ ในการแบ่งช่วงข้อมูลตามรูปแบบของ Latin Hypercubes นั้นค่าในแต่ละช่วงหาได้จาก จะนิยมแปลงให้อยู่ในช่วง  $[0, 1]$  เพื่อความสะดวก รวดเร็ว ในการคำนวณและสามารถเทียบกับค่าจริงได้ง่าย LHD สามารถสร้างได้โดยแนวคิดของการสุ่มแบบแบ่งชั้นภูมิ (Stratified Sampling) (McKay, Beckman and Conover, 1979) ทำให้มั่นใจโดยทุกๆ ส่วนย่อยของ Design Space แบ่งตัวแปรเข้าในระดับที่เท่าๆ กัน มีความเป็นไปได้เท่าๆ กัน การเลือกตัวอย่างบน Latin Hypercubes เป็น

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

โดยที่  $\pi_{ij}$  เป็นส่วนของเมทริกซ์  $n \times d$  ที่ประกอบด้วยสตมภ์  $\pi_i (i = 1, 2, \dots, d)$  โดยแต่ละสตมภ์  $\pi_i (i = 1, 2, \dots, d)$  จะสุ่มเรียงสับเปลี่ยนโดยอิสระด้วย  $1, 2, \dots, n$  และ  $U_{ij}$  คือค่า  $n \times d$  สำหรับ i.i.d. uniform  $U[0, 1]$  สุ่มค่าโดยอิสระของ  $\pi_{ij}$

$X_1$	$X_2$	$X_3$	$X_4$
1	2	4	3
2	4	1	4
3	5	3	1
4	3	2	2
5	1	5	5

### 2. Hammersley Sequence Sampling (HSS)

Hammersley Sequence Sampling เตรียมการออกแบบการทดลองที่มีคลาดเคลื่อนต่ำ สำหรับการกำหนดจุด  $n$  ในมิติ  $k$  ของ hypercube (Kalagnanam and Diwekar, 1997) เตรียมคุณสมบัติความเป็นอันหนึ่งอันเดียวได้ดีกว่าบน  $k$  มิติ มากกว่า Latin Hypercubes คลาดเคลื่อนต่ำมีความหมายว่ามีความเป็นอันหนึ่งอันเดียวกันของการกระจายของจุดใน Design Space

### 3. Uniform Design (UD)

Uniform Design เตรียมจุดที่ออกแบบการกระจายอย่างเดียวกันในโดเมนของการทดลอง Uniform Design เป็นรูปแบบหนึ่งของ Fractional Factorial Design ซึ่งเพิ่มคุณสมบัติความเป็นแบบ

เดียวกัน นิยมใช้กันอย่างแพร่หลายตั้งแต่ปี 1980 (Fang, 1980) ถ้าโดเมนของการทดลองมีจำกัด Uniform Design จะเหมือน Latin Hypercubes เมื่อโดเมนของการทดลองต่อเนื่อง หลักที่แตกต่างของการออกแบบคือใน Latin Hypercubes จะใช้การสุ่มแต่ Uniform Design จะเลือกจากค่าตรงกลาง นอกจากนี้ Latin Hypercubes ต้องการความสมดุลหนึ่งมิติสำหรับทุกระดับของตัวแปร แต่ Uniform Design ต้องการความสมดุลหนึ่งมิติสำหรับทุกระดับของตัวแปรแล้วยังต้องการความเป็นหนึ่งเดียวกันใน  $n$  มิติ ดังนั้นการออกแบบจะเหมือนกันในหนึ่งมิติแต่จะยิ่งแตกต่างกันมากขึ้นเมื่อมิติมากขึ้น การสร้าง Uniform Design ทำได้ยากกว่าและมีข้อจำกัดมากกว่า LHD

โดยในงานวิจัยนี้ ผู้วิจัยจะเลือกใช้ กลาสของการวางแผนแบบ Latin Hypercubes Design (LHD) เพราะเป็นที่นิยมในงานด้าน CSE และมีความยืดหยุ่น เมื่อกำหนดมิติปัญหา  $n \times d$  ใดๆ

### กล่องดำ หรือ Black Box

Simulation Routine หรือ กล่องดำ (Black Box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดว่าเกิดอะไรขึ้นภายใน ซึ่งกระบวนการในกล่องดำ เป็นกระบวนการที่มีการทำงานซับซ้อน โดยมากจะจำลองการทำงานของระบบการทำงานหรือปรากฏการณ์อย่างใดอย่างหนึ่ง ในรูปแบบของสมการคณิตศาสตร์ที่มีความซับซ้อนและยากต่อการอธิบายลักษณะการทำงาน ในมุมมองของผู้ใช้ กล่องดำนี้จะเปรียบเสมือนกลไกที่เราไม่ทราบได้ เพื่อศึกษากลไกดังกล่าว ในกลุ่มช่วงตัวแปรที่กำหนด  $X$  เพื่อได้ค่า  $Y$  ออกมา ซึ่งผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ ของปัจจัยที่เกี่ยวข้อง ( $X$ ) ที่ป้อนเข้าไป กล่องดำจะเกิดจากผู้เชี่ยวชาญทางด้านใดด้านหนึ่งที่ศึกษาและพัฒนาขึ้น เช่น กล่องดำจำลองการทำงานของปฏิกิริยาปฏิกิริยาปฏิกิริยา กล่องดำจำลองการทำงานของเครื่องกล เป็นต้น ซึ่งเป็นระบบที่ซับซ้อนจำลองปรากฏการณ์ที่จะเกิดเมื่อใส่ค่าตัวแปรเข้าไป การประมวลผลกล่องดำในแต่ละครั้งจะต้องใช้งบประมาณและเวลามาก จึงเป็นเหตุให้ต้องมีการวางแผนการทดลองในขั้นตอนก่อนหน้าเพื่อประหยัดงบประมาณและเวลา และเพื่อนำผลลัพธ์ดังกล่าว กับตัวแปรเข้า นำมาสร้างเป็นโมเดลประมาณค่า เพื่อให้พยากรณ์ค่าคำตอบ โดยไม่ต้องไปประมวลผลกับกล่องดำอีก



### โมเดลประมาณค่า (Approximation Models)

โมเดลประมาณค่า (Approximation Models) คือการนำกลุ่มตัวแปรเข้า (Input) และค่าผลลัพธ์ (Output) ที่ได้จากกล่องดำ มาสร้างเป็นโมเดลเพื่อประมาณค่า เพื่อให้พยากรณ์ค่าคำตอบ เทคนิคที่ใช้ในการสร้างโมเดลมีด้วยกันหลายตัว ในที่นี้จะยกตัวอย่างเทคนิคที่ใช้กันทั่วไป 2 แบบ คือ Response Surfaces Methodology และ Kriging Model

### Response Surfaces Methodology (RSM)

Response Surfaces เป็นเทคนิคที่เป็นที่นิยมใช้ โดยในขั้นต้นพัฒนาขึ้นเพื่อการวิเคราะห์สำหรับการวางแผนการทดลองทางกายภาพ (Box and Wilson, 1951) ในการวางแผนการทดลองนั้น จะประกอบด้วยตัวแปรหลายตัวโดยเฉพาะการออกแบบโมเดลแบบ Polynomial Response Surface Models ที่ถูกใช้อย่างมีประสิทธิภาพสำหรับสร้างโมเดลประมาณค่า ที่มีปัจจัยในหลากหลาย หลายงานประยุกต์การออกแบบโมเดลแบบ Second-Order Polynomial Response Surface Models มีรูปแบบดังนี้

$$\hat{y} = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i < j}^k \sum_{i < j} \beta_{ij} x_i x_j$$

ซึ่งพารามิเตอร์  $\beta_0, \beta_i, \beta_{ii}, \beta_{ij}$  คำนวณได้จากการถดถอยวิธีกำลังสองน้อยที่สุด (Least Squares Regression) ส่วน  $\hat{y}$  เป็นค่าพยากรณ์จากสมการที่สร้างจากสมการถดถอยด้วยวิธีกำลังสองน้อยที่สุด จากค่าจริง  $y(x)$  คือ ค่าจริงของตัวแปรออกโดยใช้สมการ

$$\hat{\beta} = [X'X]^{-1} X'y$$

โดย  $X$  คือ เมทริกซ์การวางแผนสำหรับจุดตัวอย่าง  $X'$  คือ ทรานสโพสของ  $X$  และ  $y$  คือเวกเตอร์หลักที่ประกอบด้วยค่าตอบสนองของแต่ละจุดตัวอย่าง Polynomial Response Surface Models สร้างได้ง่ายและสะดวกเนื่องจากมีโปรแกรมสำเร็จรูปที่รองรับการทำงานของ RSM

### Kriging Model

Kriging Model เป็นเทคนิคที่ในขั้นต้นพัฒนาเพื่อประยุกต์ใช้ในงานด้านธรณีสถิติ (Geostatistics) การสร้าง Kriging Model ใช้หลักแนวคิดคือ ตัวแปรตาม ( $y$ ) สามารถเขียนอยู่ในรูปแบบฟังก์ชันโพลิโนเมียล (Polynomial Function) ของชุดตัวแปรเข้า ร่วมกับความคลาดเคลื่อนที่เกิดจากกระบวนการสุ่ม (Stochastic Process) ซึ่งแทนด้วย  $Z(x)$  ความคลาดเคลื่อนนี้มีค่าเฉลี่ยเท่ากับ 0 และความแปรปรวนร่วมระหว่าง  $Z(u)$  และ  $Z(v)$  ใดๆ ในกระบวนการสุ่มมีค่าเท่ากับ  $V(u,v) = \sigma^2 R(u,v)$  เมื่อ  $\sigma^2$  แทนความแปรปรวนของระบบและ  $R(u,v)$  คือฟังก์ชันสหสัมพันธ์ (Correlation Function) และสามารถแทนความสัมพันธ์ด้วยรูปแบบต่าง ๆ ซึ่งรูปแบบของฟังก์ชันสหสัมพันธ์เหล่านี้สามารถดูได้จาก Koehler and Owen (1996) โดยทั่วไปแล้วตัวแปรออก  $y$  สามารถเขียนในรูปแบบทางคณิตศาสตร์ได้ดังนี้

$$y = \sum_{i=1}^k \beta_i f_i(x) + Z(x)$$

โดย  $Z(x)$  ถูกสมมติให้เป็นจริงของกระบวนการสุ่มด้วยค่ากลางศูนย์และฟังก์ชันความสัมพันธ์เชิงระยะทาง (Spatial Correlation Function) ตามสมการ

$$\text{Cov}[Z(x_i), Z(x_j)] = \sigma^2 R(x_i, x_j)$$

โดย  $\sigma^2$  คือ ความแปรปรวนของกระบวนการ ฟังก์ชันความสัมพันธ์สามารถเลือกใช้ได้หลากหลายวิธี  
อย่างไรก็ตามวิธีที่เป็นที่นิยม คือ Gaussian Correlation (Sacks, et al, 1989) รายละเอียดการสร้าง  
Kriging Model ได้อธิบายไว้ในบทที่ 3

### อัลกอริทึมการสืบค้นและเกณฑ์การเลือกค่าที่เหมาะสม

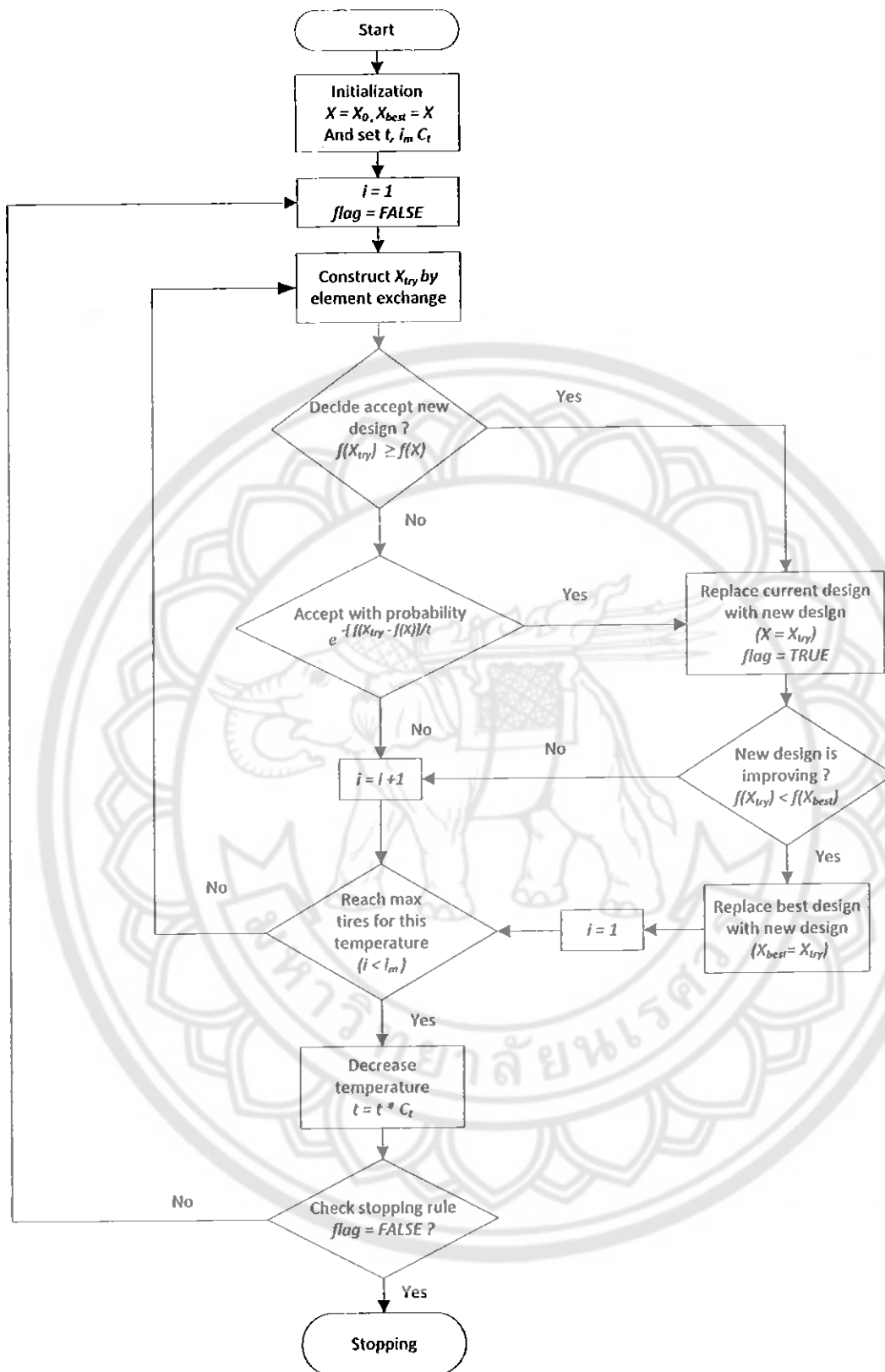
การออกแบบการทดลองที่ดีจะส่งผลให้ได้โมเดลจำลองที่มีความแม่นยำและมีประสิทธิภาพใน  
การพยากรณ์ การออกแบบการทดลองนั้นจะจำกัดด้วยจำนวนตัวแปรและจำนวนครั้งในการรัน เพราะ  
การทำการทดลองในแต่ละครั้งจะใช้เวลาและเงินเป็นจำนวนมาก การออกแบบการทดลองที่ดีที่สุดจะมา  
จาก "การเลือกทางเลือกที่ดีที่สุด" ซึ่งจะเรียกว่า การทำงานแบบการหาค่าเหมาะสม (Optimization)

ในงานวิจัยนี้ได้นำอัลกอริทึมการสืบค้นมาใช้ในการเลือกแผนการทดลองที่ดีที่สุด เพราะจำนวน  
แผนการทดลองที่เป็นไปได้มีจำนวนมหาศาล เป็นไปไม่ได้ที่จะนำทุกๆ แผนการทดลองมาทำการทดลอง  
จริง แผนการทดลองที่ได้จากอัลกอริทึมการสืบค้น จะเป็นแผนการทดลองที่เหมาะสมที่สุด เพื่อใช้เป็น  
ตัวแทนของแผนการทดลองทั้งหมดให้ได้ดี อัลกอริทึมการสืบค้นที่นิยมกันในปัจจุบันนั้นมีหลายแบบ  
เช่น Simulated Annealing: SA, Genetic Algorithm: GA, Tabu Search: TA เป็นต้น โดยอัลกอริทึม  
การสืบค้นจะต้องทำงานควบคู่ไปกับเกณฑ์การเลือกแผนการทดลองที่เหมาะสม (Optimality Criteria)  
ในที่นี้จะกล่าวถึงสองอัลกอริทึมการสืบค้น คือ อัลกอริทึมเจเนติก (Genetic Algorithm: GA) และ  
อัลกอริทึมซิมมูลเตดเทคแอนนิง (Simulated Annealing :SA)

### อัลกอริทึมการสืบค้นแบบสโตแคสติกเชิงวิวัฒนาการ

อัลกอริทึมการสืบค้นแบบสโตแคสติกเชิงวิวัฒนาการที่เป็นพื้นฐานในการวิจัยครั้งนี้ ใช้สำหรับสร้าง  
แผนการทดลองที่เหมาะสม ซึ่งอัลกอริทึมสำหรับการออกแบบการทดลองส่วนใหญ่จะมีขั้นตอนการ  
ทำงานแบบวนซ้ำ ตัวอย่างเช่น Taboo Search และ Simulated annealing เป็นต้น

Simulated annealing (SA) นำเสนอโดย Kirkpatrick, et al. (1983) เป็นอัลกอริทึมที่จำลองมา  
จากกระบวนการหลอมโลหะโดยเริ่มจากอุณหภูมิที่สูง และลดอุณหภูมิลงอย่างช้าๆ จนได้โลหะที่  
เหมาะสม ผลเฉลยของปัญหาคือสถานะของโลหะ ค่าของพลังงานเปรียบเหมือนค่าของฟังก์ชัน  
วัตถุประสงค์ที่คำนวณได้จากผลเฉลย SA ถูกนำมาใช้ในการค้นหาแผนการทดลองสำหรับ CSE ครั้ง  
แรกโดย Morris and Mitchell (1995) ซึ่งมีขั้นตอนการทำงานดังภาพในหน้าถัดไป ซึ่งสามารถอธิบาย  
ดังนี้



Simulated annealing flowchart

ขั้นตอนที่ 1 สร้างแผนการทดลองเริ่มต้น  $X_0$  และตั้งค่าพารามิเตอร์เริ่มต้นได้แก่ อุณหภูมิ  $t$  ตามสมการค่าเฉลี่ยของระยะทางระหว่างจุดทุกคู่ของแผนการทดลอง  $X_0$  ดังนี้

$$t = t_0 = \sum_{i=1}^n \sum_{i=i+1}^n \left[ \sum_{k=1}^d (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}} / \binom{d}{2}$$

จำนวนรอบของการสร้างแผนการทดลอง  $i_m$  อัตราส่วนสำหรับลดอุณหภูมิ  $C_t$  และให้  $X = X_0$  และ

$$X_{best} = X$$

ขั้นตอนที่ 2 ตั้งค่าพารามิเตอร์สำหรับตรวจสอบการหยุดสืบค้น  $flag = FALSE$  และ  $i = 1$

ขั้นตอนที่ 3 สร้างแผนการทดลองใหม่  $X_{try}$  โดยการแลกเปลี่ยนสมาชิกในคอลัมน์

$$X_{ak} \leftrightarrow X_{bk} \text{ (อธิบายอย่างละเอียดในบทที่ 3)}$$

ขั้นตอนที่ 4 คำนวณค่า  $\Phi_p$  ของ  $X_{try}$  และแผนการทดลองปัจจุบัน  $X$  เพื่อตัดสินใจยอมรับแผนการทดลองใหม่ถ้าเป็นไปตามเงื่อนไข  $\Phi_p(X_{try}) < \Phi_p(X)$  หรือยอมรับตามความน่าจะเป็นตามสมการ (6)

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

เมื่อ  $t$  คืออุณหภูมิขณะนั้น และถ้า  $X_{try}$  ถูกยอมรับ ให้  $X = X_{try}$  และให้  $flag = TRUE$

ขั้นตอนที่ 5 ตรวจสอบกระบวนการค้นหาในรอบนั้นว่าแผนการทดลองที่ได้คุณสมบัติที่ดีที่สุดหรือไม่โดยเปรียบเทียบแผนการทดลองใหม่  $X_{try}$  กับแผนการทดลองที่ดีที่สุด  $X_{best}$  ถ้า  $\Phi_p(X_{try}) < \Phi_p(X_{best})$  ให้ทำการแทนที่  $X_{best} = X_{try}$

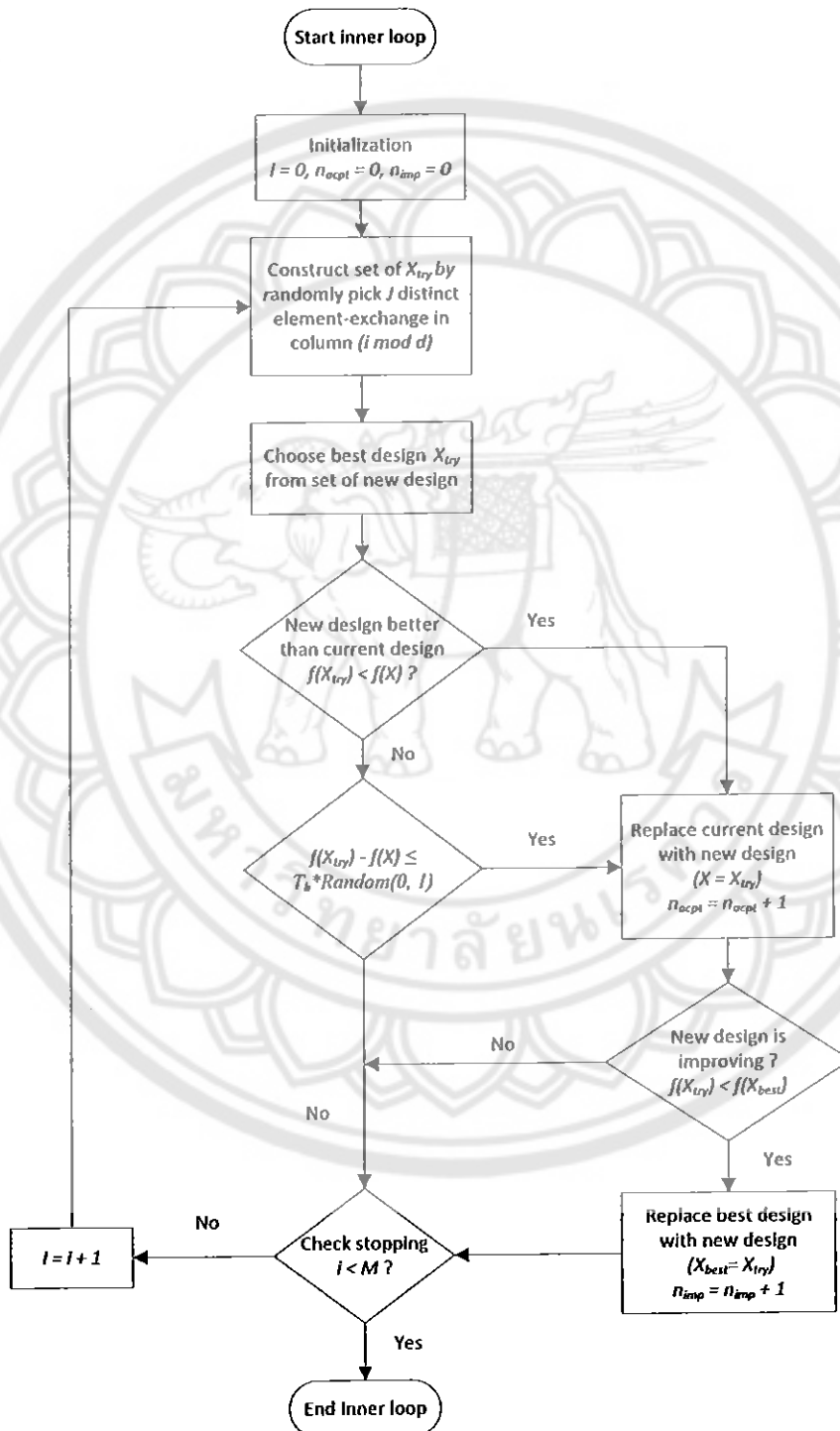
ขั้นตอนที่ 6 ทำงานวนซ้ำกลับไปขั้นตอนที่ 3 จนกว่าจะครบตามจำนวนรอบ  $i_m$  แล้วจึงทำการลดอุณหภูมิลงด้วยสมการ  $t = t * C_t$

ขั้นตอนที่ 7 ทำงานวนซ้ำกลับไปขั้นตอน 2 จนกว่าจะเป็นไปตามเงื่อนไขการหยุดสืบค้นคือ  $flag = FALSE$  เป็นจริง

**Stochastic Evolutionary (SE)** นำเสนอโดย Sabb and Rao (1991) ได้แรงบันดาลใจจากวิวัฒนาการของสิ่งมีชีวิต โดยหวังว่าวิวัฒนาการจะช่วยกำจัดลักษณะที่ไม่ดีในรุ่นเก่า และได้ลักษณะที่ดีกว่าในรุ่นถัดไป การยอมรับผลเฉลยใหม่ของ SE อยู่บนพื้นฐานของพารามิเตอร์ควบคุม (Threshold:  $T_h$ ) ซึ่ง SE สามารถหาผลเฉลยของปัญหา TSP ได้ในเวลาดีกว่า SA แต่ปัญหาของ SE อยู่ที่การตั้งค่า  $T_h$  สำหรับปัญหาที่ต่างออกไป และ SE ถูกพัฒนาให้สร้างแผนการทดลองสำหรับการออกแบบการ



ทดลองด้วยคอมพิวเตอร์โดย Jin, et al. (2005) และให้ชื่ออัลกอริทึมใหม่ว่า Enhanced stochastic evolutionary (ESE) ซึ่ง ESE ได้ขจัดข้อเสียของ SE โดยการรวมประสบการณ์ที่ผ่านมาของอัลกอริทึมมาใช้ควบคุม  $T_h$  ทำให้ค่า  $T_h$  สามารถปรับตัวเองได้เพื่อให้เข้ากับปัญหาหรือเงื่อนไขที่ต่างออกไป ESE มีการทำงานแบบวนซ้ำ (Loop) ซ้อนกัน 2 ลูป (เรียกว่าลูปใน และลูปนอก) ดังแสดงโดย 2 ภาพต่อไปนี้ โดยสามารถอธิบายขั้นตอนการทำงานได้ดังนี้



ESE (Inner loop) flowchart ที่มา: Jin, et al. (2005)



15 10669x

### กลไกของลูปใน (Inner loop)

การทำงานของลูปในเป็นกระบวนการวนซ้ำทั้งหมด  $M$  รอบทำหน้าที่สร้างกลุ่มของแผนการทดลองใหม่  $X_{try}$  โดยการสุ่มแลกเปลี่ยนสมาชิก ( $X_{aj} \leftrightarrow X_{bj}$ ) และตัดสินใจยอมรับแผนการทดลองใหม่

22 ส.ค. 2554

ขั้นตอนที่ 1 รับแผนการทดลองเริ่มต้น  $X_0$  จากลูปนอก และให้ค่า  $i = 0, n_{acpt} = 0$  และ  $n_{acpt} = 0$

ขั้นตอนที่ 2 สร้างกลุ่มของแผนการทดลอง  $X_{try}$  จำนวน  $J$  แบบที่ไม่ซ้ำกันโดยการแลกเปลี่ยนสมาชิกในคอลัมน์ที่  $i \bmod d$  (จำนวนตัวแปรในปฏิภูมิการทดลอง)

ขั้นตอนที่ 3 คำนวณค่าฟังก์ชันวัตถุประสงค์  $\Phi_p$  ทั้งหมดของกลุ่มแผนการทดลอง  $X_{try}$  และเลือกแผนการทดลอง  $X_{try}$  ที่ดีที่สุดจากในกลุ่ม

ขั้นตอนที่ 4 ตัดสินใจยอมรับแผนการทดลอง  $X_{try}$  ถ้า  $\Phi_p(X_{try}) < \Phi_p(X)$  หรือยอมรับด้วยความน่าจะเป็นตามสมการ

$$\Phi_p(X_{try}) - \Phi_p(X) \leq T_h * random(0,1)$$

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

และเพิ่มค่า  $n_{acpt} = n_{acpt} + 1$  เมื่อแผนการทดลอง  $X_{try}$  ถูกยอมรับ

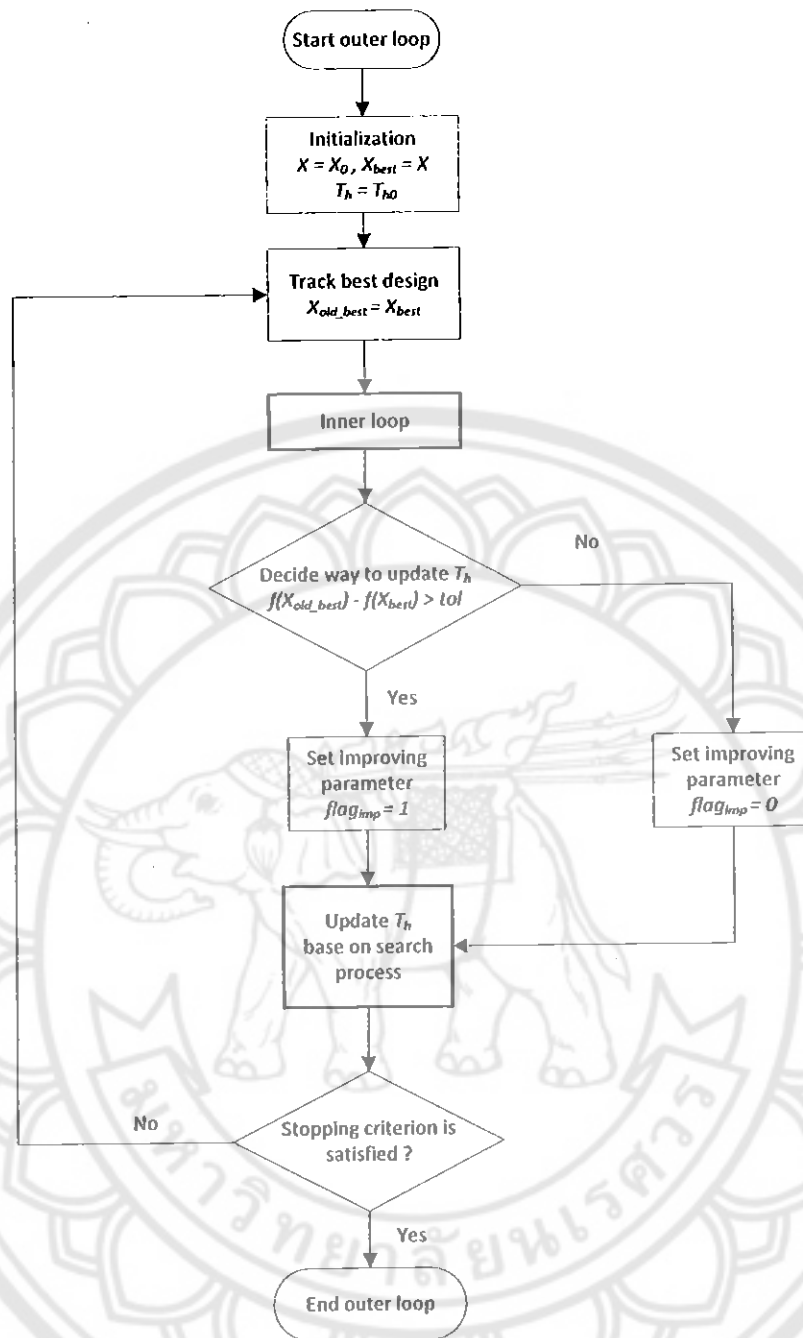
ขั้นตอนที่ 5 เปรียบเทียบแผนการทดลอง  $X_{try}$  กับแผนการทดลอง  $X_{best}$  ถ้า

$\Phi_p(X_{try}) < \Phi_p(X_{best})$  ให้ทำการแทนที่  $X_{best} = X_{try}$  และเพิ่มค่า  $n_{imp} = n_{imp} + 1$

ขั้นตอนที่ 6 ตรวจสอบเงื่อนไขการหยุดถ้า  $i < M$  เป็นจริง โดยที่  $M = 2 \binom{n}{2} d/J$  แต่ไม่เกิน 100

### กลไกของลูปนอก (Outer loop)

การทำงานของลูปนอก ทำหน้าที่ควบคุมกระบวนการค้นหาแบบเชิงกลุ่ม(Global search) โดยการปรับค่าพารามิเตอร์ควบคุม  $T_h$  เริ่มต้นการทำงานของอัลกอริธึมค่า  $T_h$  จะถูกตั้งให้ไว้ต่ำ (จะกล่าวถึงในส่วนของขั้นตอนการทำงาน) หลังจากนั้นจะถูกปรับค่าไปตามกระบวนการค้นหาซึ่งแบ่งได้เป็น 2 กรณี คือ ถ้าแผนการทดลองที่ได้หลังจากจบการทำงานของลูปในดีขึ้นจะใช้วิธีการ การค้นหาแบบปรับ (Improving process) เพื่อให้แผนการทดลองที่ได้มีคุณสมบัติดียิ่งขึ้น และถ้าแผนการทดลองที่ได้หลังจากจบการทำงานของลูปในไม่ดีขึ้นจะเปลี่ยนไปใช้ การค้นหาแบบสำรวจ (Exploration process) เพื่อหลีกเลี่ยงจากแผนการทดลองที่เหมาะสมเฉพาะที่



ESE (Outer loop) flowchart ที่มา: Jin, et al. (2005)

ขั้นตอนที่ 1 สร้างแผนการทดลองเริ่มต้น  $X_0$  ให้ค่าเริ่มต้น  $X = X_0$ ,  $X_{best} = X$  และ  $T_h$

ตามสมการ

$$T_h = T_{h0} = 0.005 * \Phi_p(X_0)$$

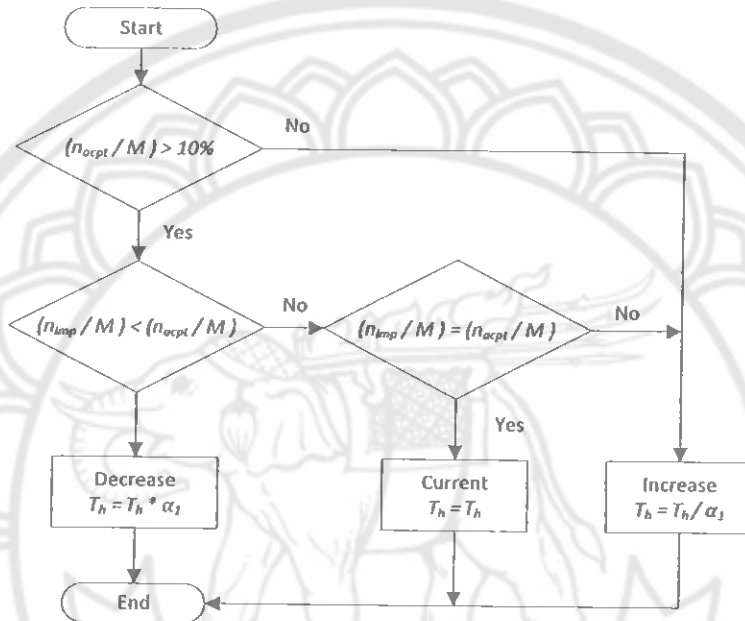
ขั้นตอนที่ 2 บันทึกแผนการทดลอง  $X_{old\_best} = X_{best}$

ขั้นตอนที่ 3 เข้าสู่การทำงานในส่วนของลูปใน

ขั้นตอนที่ 4 เลือกกระบวนการค้นหาถ้า  $\Phi_p(X_{old\_best}) - \Phi_p(X_{best}) > tol$  ให้  $flag_{imp} = 1$  ถ้าไม่ใช่ให้  $flag_{imp} = 0$   
 ขั้นตอนที่ 5 ปรับค่า  $T_h$  ตามลักษณะกระบวนการค้นหา (อธิบายในส่วนถัดไป)  
 ขั้นตอนที่ 6 ตรวจสอบกฎการหยุดสืบทัน ถ้าไม่เป็นไปตามเงื่อนไข ให้วนกลับไปยังขั้นตอนที่

2

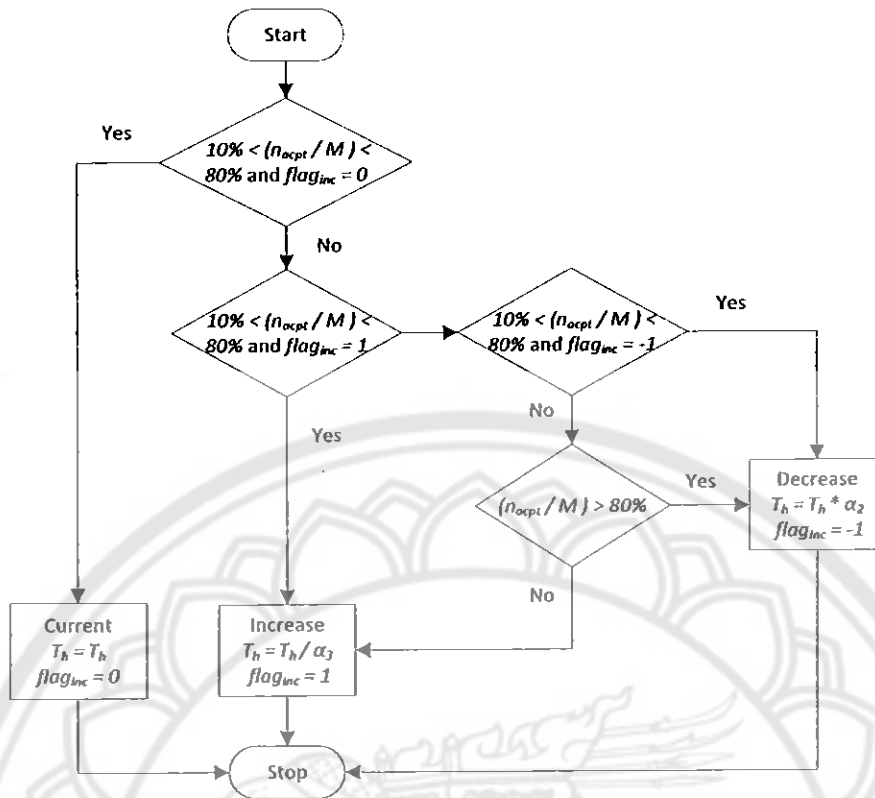
การปรับค่า  $T_h$  ในขั้นตอนที่ 5 ของรูปนอกแบ่งได้ 2 กรณีคือ การค้นหาแบบปรับ (Improving process) เมื่อ  $flag_{imp} = 1$  และการค้นหาแบบสำรวจ (Exploration process) เมื่อ  $flag_{imp} = 0$



Improving process flowchart

การค้นหาแบบปรับ (Improving process) ค่า  $T_h$  จะถูกปรับเพื่อช่วยให้อัลกอริทึมสามารถค้นหาแผนการทดลองที่มีคุณสมบัติดีขึ้น ( $\Phi_p$  น้อยลง) มีพื้นฐานบน  $n_{acpt}/M$  (อัตราส่วนของการยอมรับ  $X_{try}$  กับจำนวนรอบการทำงานลูปใน) และ  $n_{imp}/M$  (อัตราส่วนของการได้  $X_{try}$  ที่มีคุณสมบัติดีขึ้น กับจำนวนรอบการทำงานลูปใน) ตามเงื่อนไข (ภาพ 7) ดังนี้

1.  $T_h$  ลดลงตามสมการ  $T_h = T_h * \alpha_1$  ถ้า  $n_{acpt}/M > 10\%$  และ  $n_{imp}/M < n_{acpt}/M$
2.  $T_h$  ไม่เปลี่ยนแปลงถ้า  $n_{acpt}/M > 10\%$  และ  $n_{imp}/M = n_{acpt}/M$
3. นอกเหนือจาก 2 เงื่อนไขแรกให้ทำการเพิ่ม  $T_h$  ตามสมการ  $T_h = T_h / \alpha_2$  โดยกำหนดให้  $\alpha_1 = 0.8$  และ  $0 < \alpha_1 < 1$  (Jin, et al., 2005)



Exploration process flowchart

การค้นหาแบบสำรวจ (Exploration process) ค่า  $T_h$  ถูกปรับเพื่อช่วยให้อัลกอริทึมหนีจากแผนการทดลองที่เหมาะสมแบบเฉพาะที่โดยพิจารณาจาก  $n_{acpt}/M$  ตามเงื่อนไข (ภาพ 8)

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  $0 < \alpha_2 < \alpha_3 < 1$  และกระบวนการนี้จะวนซ้ำ (Repeat) จนกว่ากระบวนการค้นหาจะพบแผนการทดลองที่มีคุณสมบัติดีขึ้น

### ทฤษฎีความซับซ้อนในการคำนวณ

การวิเคราะห์หาความซับซ้อนด้านเวลา (Time complexity) สำหรับอัลกอริทึมที่ใช้ในการแก้ปัญหาใดปัญหาหนึ่ง เมื่อมีการทำงาน (Execute) บนเครื่องคอมพิวเตอร์แต่ละเครื่องที่แตกต่างกันเวลาที่ใช้จะแตกต่างกันตามไปด้วย เพื่อหลีกเลี่ยงความแตกต่างในจุดนี้ สัญลักษณ์โอใหญ่ (Big O) ถูกนำมาใช้บอกถึงขอบบน (Upper bound) ของฟังก์ชันการเติบโตทางเวลาที่ใช้ในการทำงานของอัลกอริทึมโดยประมาณ ในกรณีที่แย่ที่สุด (Worst case) ซึ่งเป็นความสัมพันธ์ระหว่างเวลาที่กล่าวไปในข้างต้นกับขนาดของตัวแปรเข้า ซึ่งมีหลักการพิจารณาคือ  $f(n)$  เป็น  $O(g(n))$  เมื่อ สามารถหาค่าคงที่  $c \in I^+$  ที่ทำให้  $f(n) \leq cg(n)$  เมื่อ  $n \geq n_0$

## เกณฑ์การเลือกค่าที่เหมาะสม

จากอัลกอริทึมที่อธิบายไว้ในข้างต้นนั้นหลักการทำงานของอัลกอริทึมการสืบค้นต้องทำงานร่วมกับเกณฑ์การเลือกค่าที่เหมาะสม (Optimality Criteria) เสมอ ในการค้นหาแผนการทดลองที่ดีที่สุดนั้น ผู้วิจัยหลายท่านได้เสนอเกณฑ์การเลือกไว้หลายเกณฑ์ เช่น Maximin Distance Criterion, Entropy Criterion และ Centered  $L_2$  Discrepancy Criterion โดยในการวิจัยนี้จะเลือกใช้ Maximin Distance Criterion และ  $\phi_p$  Criterion มาเป็นเกณฑ์ ทั้งนี้เพราะลักษณะเกณฑ์ทั้งสองตัวนี้มีคุณสมบัติเด่น คือ ไม่ขึ้นต่อโมเดล (Model Independent) ทำให้สะดวกในการวางแผนการทดลอง เพื่อนำไปใช้กับการทดลองต่างๆ ไปได้

### แม็กซิมิน (Maximin)

โดยหลักการของเกณฑ์การเลือกค่าที่เหมาะสม แบบแม็กซิมิน (Maximin) นั้นจะต้องหา ระยะห่างจุดทดลองบนแผนการทดลองก่อน การหาระยะห่างของจุดนั้นหาได้สองแบบคือ Rectangular Distance และ Euclidean Distance ดังสมการ

$$d(x_s, x_t) = \sum_{i=1}^k |x_{si} - x_{ti}| \text{ และ } d(x_s, x_t) = \left[ \sum_{i=1}^k (x_{si} - x_{ti})^2 \right]^{1/2} \text{ ตามลำดับ}$$

งานวิจัยนี้ได้เลือกใช้ Euclidean Distance โดยจากแผนการทดลองขนาด  $n \times d$  เมื่อนำมาหาระยะห่างจุดจะได้เป็นเมทริกซ์ระยะห่างจุดขนาด  $n \times n$  เป็นเมทริกซ์แบบสมมาตรเส้นทแยงมุมจะมีค่าเป็น 0 ค่าระยะห่างที่น้อยที่สุดของเมทริกซ์ระยะห่างนี้ จะนำมาเป็นตัวเปรียบเทียบกันกับ เมทริกซ์ของแผนการทดลองอื่นๆ คือ ค่าที่น้อยที่สุดของเมทริกซ์ระยะห่างของแผนการทดลองใดมีค่ามากกว่ากัน (Maximin Design) จะเลือกแผนการทดลองนั้น ตัวอย่างเช่น แผนการทดลองขนาด  $9 \times 2$

0.375	0.875
0.125	0.375
0.625	0.125
0.250	0
1	0.250
0.875	0.625
0.500	0.500
0	0.75
0.75	1

หาค่าระยะห่างโดยเลือกใช้ Euclidean Distance ได้ดังนี้

$$d_{11} = \left( (0.375 - 0.375)^2 + (0.875 - 0.875)^2 \right) = 0$$

$$d_{12} = \left( (0.375 - 0.125)^2 + (0.875 - 0.375)^2 \right) = 0.5590$$

$$d_{21} = \left( (0.125 - 0.375)^2 + (0.375 - 0.875)^2 \right) = 0.5590$$

⋮

$$d_{98} = ((0.75 - 0)^2 + (1 - 0.75)^2) = 0.7906$$

$$d_{99} = ((0.75 - 0.75)^2 + (1 - 1)^2) = 0$$

ซึ่งได้เป็นเมทริกซ์ระยะห่าง ดังนี้

0	0.5590	0.7906	0.8839	0.8839	0.5590	0.3953	0.3953	0.3953
0.5590	0	0.5590	0.3953	0.8839	0.7906	0.3953	0.3953	0.8839
0.7906	0.5590	0	0.3953	0.3953	0.5590	0.3953	0.8839	0.8839
0.8839	0.3953	0.3953	0	0.7906	0.8839	0.5590	0.7906	1.1180
0.8839	0.8839	0.3953	0.7906	0	0.3953	0.5590	1.1180	0.7906
0.5590	0.7906	0.5590	0.8839	0.3953	0	0.3953	0.8839	0.3953
0.3953	0.3953	0.3953	0.5590	0.5590	0.3953	0	0.5590	0.5590
0.3953	0.3953	0.8839	0.7906	1.1180	0.8839	0.5590	0	0.7906
0.3953	0.8839	0.8839	1.1180	0.7906	0.3953	0.5590	0.7906	0

จากเมทริกซ์ระยะห่าง ต้องหาค่าที่น้อยที่สุด จากตัวอย่างคือค่า 0.3953 ค่านี้จะเป็นตัวเปรียบเทียบกับค่าระยะห่างที่น้อยที่สุดของแผนการทดลองอื่นๆ โดยจะพิจารณาว่าค่ามากกว่าจะเป็นแผนการทดลองที่เหมาะสมกว่า

ฟายฟี ( $\phi_p$ )

โดยหลักการของเกณฑ์การเลือกค่าที่เหมาะสม แบบ  $\phi_p$  นั้นเป็นวิธีที่พัฒนาจาก Maximin Distance Criterion นำเสนอโดย Morris และ Mitchell (1993) จะหาค่าได้จากสูตร

$$\phi_p = \left[ \sum_{j=1}^m J_j d_j^{-p} \right]^{1/p}$$

โดย  $d_j$  คือ ลิสต์ของระยะห่าง และ  $J_j$  คือ ลิสต์ของดัชนี ซึ่งเป็นจำนวนคู่ของจุดทดลองที่ถูกแยกด้วยระยะ  $d_j$  และ  $p$  เป็นเลขจำนวนเต็มบวก ซึ่งต้องมีค่ามากพอ จากงานวิจัยก่อนหน้าค่า  $p$  ที่มากพอคือ 5 จากตัวอย่างแผนการทดลองข้างต้น เมื่อหาระยะห่างจุดได้เป็นเมทริกซ์ระยะห่างขนาด  $n \times n$  คือ  $(9 \times 9)$  แล้ว ต้องนำมาเรียงลำดับจากน้อยไปมาก แต่ก่อนจะเรียงลำดับ จะเห็นว่าเป็นเมทริกซ์ที่ได้นั้นเป็นเมทริกซ์สมมาตร (Symmetric Matrix) ซึ่งจะคำนวณเฉพาะส่วนด้านบนของเส้นทแยง เพื่อลดเวลาในการคำนวณ โดยที่ความถูกต้องของคำตอบยังเหมือนกัน (Morris and Mitchell, 1995) ดังนั้นเมทริกซ์ที่จะพิจารณาเป็นดังนี้

0	0.5590	0.7906	0.8839	0.8839	0.5590	0.3953	0.3953	0.3953
0	0	0.5590	0.3953	0.8839	0.7906	0.3953	0.3953	0.8839
0	0	0	0.3953	0.3953	0.5590	0.3953	0.8839	0.8839
0	0	0	0	0.7906	0.8839	0.5590	0.7906	1.1180
0	0	0	0	0	0.3953	0.5590	1.1180	0.7906
0	0	0	0	0	0	0.3953	0.8839	0.3953
0	0	0	0	0	0	0	0.5590	0.5590
0	0	0	0	0	0	0	0	0.7906
0	0	0	0	0	0	0	0	0

นำมาเรียงระยะห่างน้อยไปมากจะได้ลิสต์ของระยะห่าง  $(d_1, d_2, \dots, d_m)$  ซึ่งเป็นเฉพาะส่วนบนของเส้นทแยงมุมของเมทริกซ์ระยะห่าง จะมีจำนวน  $m$  เป็นค่าตั้งแต่ 1 ถึง  $\binom{n}{2}$  และได้ลิสต์ของดัชนี  $(J_1, J_2, \dots, J_m)$  โดย  $J_i$  ลิสต์ของดัชนี ซึ่งเป็นจำนวนคู่ของจุดทดลองที่ถูกแยกด้วยระยะ  $d_i$  จากตัวอย่างจะได้ ลิสต์ของระยะห่างเป็น

$$(0.3953, 0.5590, 0.7906, 0.8839, 1.1180)$$

ลิสต์ของดัชนี  $(J_1, J_2, \dots, J_m)$  เป็น

$$(12, 8, 6, 8, 2)$$

แล้วนำค่า  $d_i$  และ  $J_i$  มาคำนวณตามสูตรดังกล่าวไปแล้ว โดยค่า  $\phi_p$  มีค่าน้อยแสดงว่าแผนการทดลองนั้นเป็นแผนการทดลองที่ดี

### สถิติที่เกี่ยวข้อง

เกณฑ์ที่ใช้วัดความแม่นยำในการพยากรณ์ คือ Root Mean Square Error: RMSE จะใช้วัดค่าความแม่นยำให้แผนการทดลองที่เกิดจากอัลกอริทึม SA กับ แผนการทดลองที่ได้จาก DACE MATLAB Toolbox ซึ่งเป็น Kriging Model คำนวณได้ตามสูตร

$$RMSE = \sqrt{\frac{\sum_{i=1}^k (y_i' - \hat{y}_i')^2}{k}}$$

โดยที่  $y_i'$  คือ ผลลัพธ์ที่เกิดจากกล่องดำ,  $\hat{y}_i'$  คือ ค่าพยากรณ์ที่ได้จาก Kriging Model,  $k$  คือ จำนวนจุดทดสอบ





### บทที่ 3

## วิธีดำเนินการวิจัย

ในบทนี้ผู้วิจัยจะกล่าวถึงวิธีการพัฒนาอัลกอริทึมการสับค่านแบบสโทแคสติกเชิงวิวัฒนาการ โดยใช้ภาษา R ซึ่งประกอบด้วย การแลกเปลี่ยนสมาชิก การปรับปรุงอัลกอริทึมการสับค่าน การประเมินค่า  $\Phi_p$  แบบใหม่ และการวิเคราะห์กฎการหยุดสับค่านที่เหมาะสม

#### 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  $\Phi_p$  แบบใหม่สำหรับแผนการทดลอง  $X_{try}$  ได้อย่างมีประสิทธิภาพ (Jin, et al., 2005) ตัวอย่างการแลกเปลี่ยนสมาชิกแสดงในภาพ

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

#### การแลกเปลี่ยนสมาชิกของแผนการทดลองแบบ LHD ขนาด 9 x 2

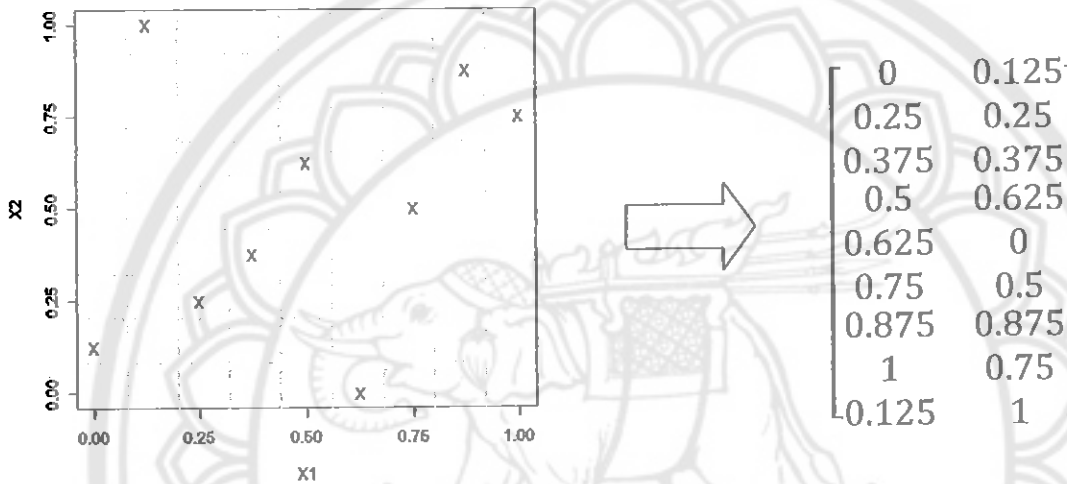
จากภาพแสดงการแลกเปลี่ยนสมาชิกโดยมี  $k = 2, a = 3, b = 6$  จากแผนการทดลองเริ่มต้น (ด้านซ้ายมือ) สมาชิกในคอลัมน์ที่ 2 แถวที่ 3 คือ 0.375 จะถูกสลับที่กับสมาชิกในคอลัมน์ที่ 2 แถวที่ 6 คือ 0.5 เปลี่ยนไปเป็นแผนการทดลองใหม่ (ด้านขวามือ)

## การเพิ่มประสิทธิภาพอัลกอริธึมการสืบค้น

การนำทฤษฎีที่มีอยู่มาปรับใช้เพื่อเพิ่มประสิทธิภาพของอัลกอริธึมการสืบค้น เช่นการปรับปรุงวิธีการค้นหาแบบเชิงกลุ่มเพื่อให้การค้นหาแผนการทดลองสำหรับ CSE มีประสิทธิภาพมากขึ้น

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

การพัฒนาอัลกอริธึม Modified simulated annealing (MSA) ขึ้นโดยใช้ตัวแปลภาษา R เพื่อใช้ในการค้นหาแผนการทดลองสำหรับ CSE ดังภาพซึ่งสามารถอธิบายขั้นตอนการทำงานและวิธีการตั้งค่าพารามิเตอร์เริ่มต้นดังนี้



การสร้างแผนการทดลองแบบ LHD เริ่มต้นขนาด 9 x 2

ขั้นตอนที่ 1 สร้างแผนการทดลองเริ่มต้น  $X_0$  โดยการสุ่มเรียงสับเปลี่ยนค่าที่เป็นไปได้ของตัวแปรในแผนการทดลองตามหลักการของ LHD ดังภาพ 10 ให้ค่ากับแผนการทดลองปัจจุบัน  $X$  และทำการบันทึกแผนการทดลอง  $X$  ให้เป็นแผนการทดลองที่ดีที่สุด (Best so far)  $X_{best} = X$  เพื่อใช้สำหรับเปรียบเทียบกับแผนการทดลองใหม่  $X_{try}$  และตั้งค่าพารามิเตอร์เริ่มต้นได้แก่ อุณหภูมิ  $t$  ที่เป็นเสมือนพารามิเตอร์ควบคุมตามสมการค่าเฉลี่ยของระยะทางระหว่างจุดทุกคู่ของแผนการทดลอง  $X_0$  ดังนี้

$$t = t_0 = \sum_{i=1}^n \sum_{i=i+1}^n \left[ \sum_{k=1}^d (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}} / \binom{d}{2}$$

จำนวนรอบการสร้างแผนการทดลองใหม่  $i_m$  ซึ่งค่านี้จะต้องมากเพียงพอที่สามารถทำให้อัลกอริธึม MSA หลุดจากการหาผลเฉลยเฉพาะที่ได้ โดย  $i_m$  ถูกตั้งค่าตามจำนวนของตัวแปรเข้า ในแผนการทดลองคือ 1000 สำหรับจำนวนตัวแปรที่มากกว่าหรือเท่ากับ 6 และ 500 สำหรับจำนวนตัวแปรตั้งแต่ 7 ตัวขึ้นไป อัตราส่วนสำหรับลดอุณหภูมิ  $c_t \leq 0.95$  เพื่อให้การลดอุณหภูมิ  $t$  ในกระบวนการค้นหา

เป็นไปอย่างช้าๆ ค่าผลต่างของการยอมรับแผนการทดลองใหม่  $tol = 0.0001$  ซึ่งจากการศึกษาพบว่า ถ้า  $tol$  น้อยเกินไปจะทำให้อัลกอริทึม MSA ใช้เวลาในการทำงานนานขึ้นแต่ไม่ทำให้ผลเฉลยที่ได้ดีขึ้น

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  $\Phi_p$  ของแผนการทดลอง  $X$  และ  $X_{try}$  เพื่อตัดสินใจว่าจะยอมรับ (Accept) แผนการทดลอง  $X_{try}$  แทนที่แผนการทดลอง  $X$  หรือไม่ตามสมการ

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

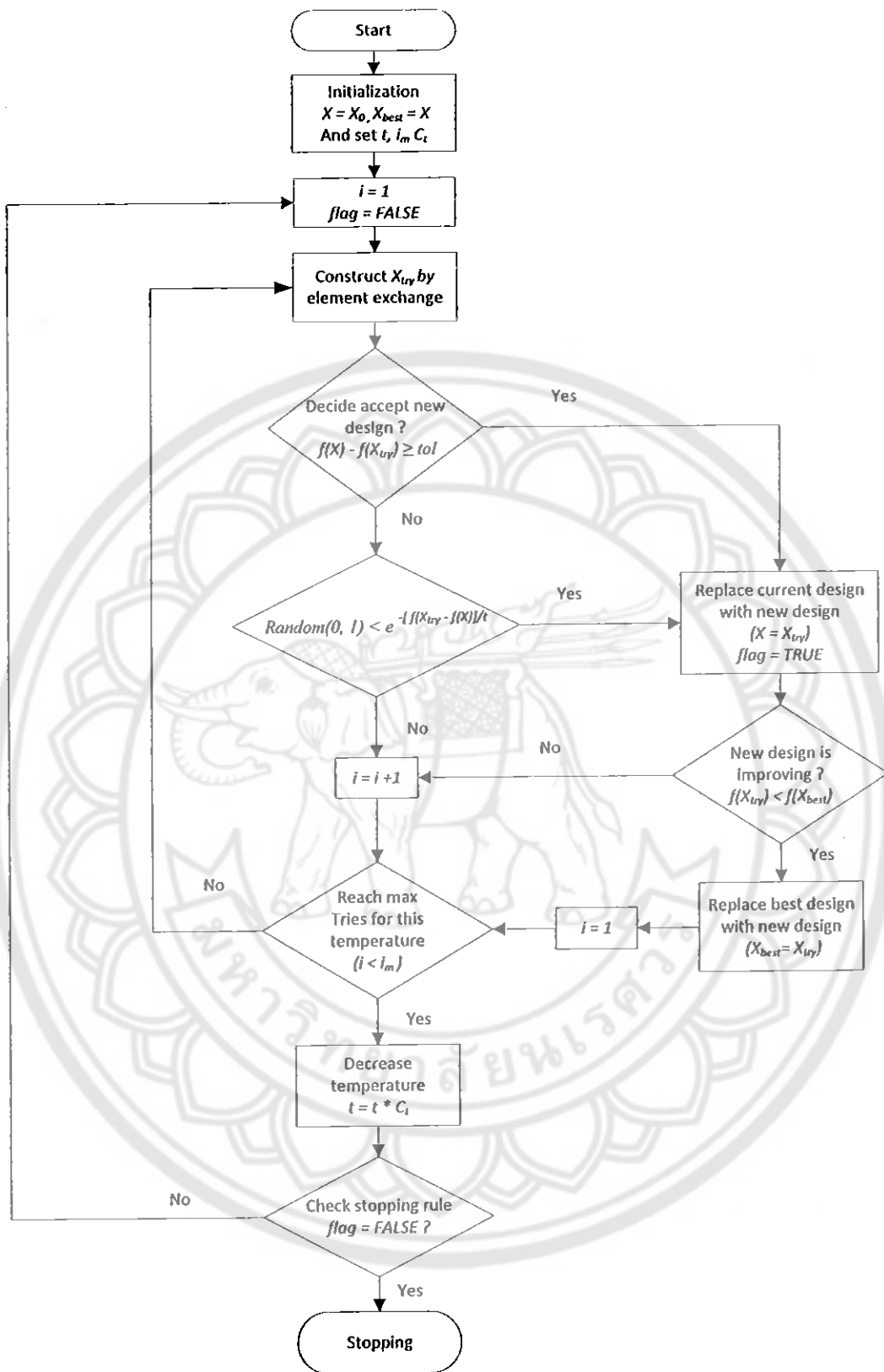
โดย  $tol = 0.0001$  หรือยอมรับแผนการทดลอง  $X_{try}$  ตามความน่าจะเป็นในสมการ

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

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  $i_m$  แล้วจึงทำการลดอุณหภูมิลงด้วยสมการ  $t = t * c_t$

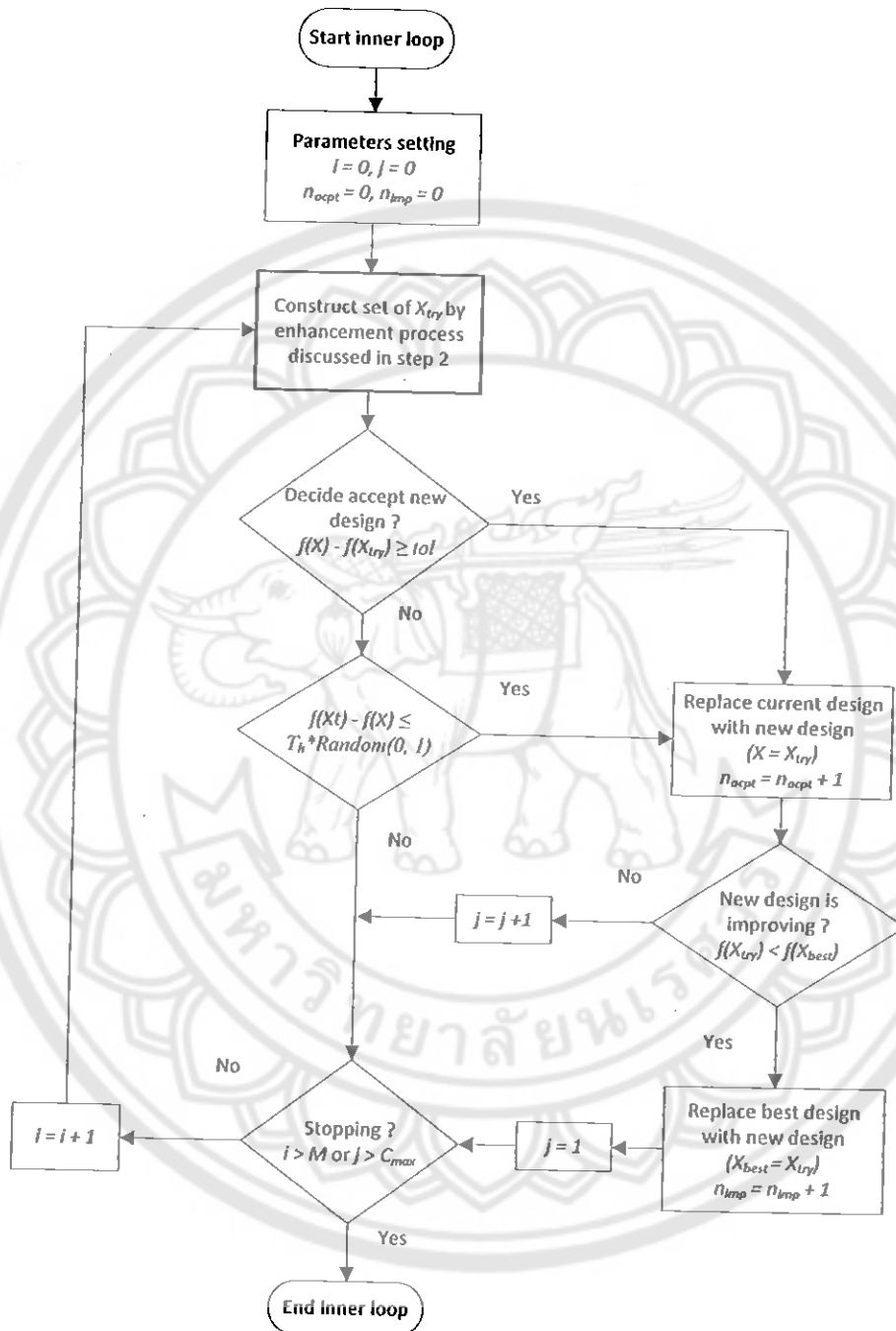
ขั้นตอนที่ 6 ขั้นตอนการทำงานจะวนซ้ำกลับไปขั้นตอนที่ 2 จนกว่าจะเป็นไปตามเงื่อนไขการหยุดทำงานของอัลกอริทึม  $flag = FALSE$  เป็นจริง



**Modified simulated annealing algorithm flowchart**

## 2. Enhancement of enhanced stochastic evolutionary

อัลกอริทึม Enhancement of enhanced stochastic evolutionary พัฒนาขึ้นโดยใช้ตัวแปรภาษา R เช่นเดียวกับ MSA



Enhancement of enhanced stochastic evolutionary flowchart (รูปใน)

ผู้วิจัยได้พัฒนา EESE จากการนำเอาข้อดีของอัลกอริทึม MSA ที่นำเสนอโดย Thamma, et al. (2008) คือ ส่วนของวิธีการค้นหาแบบเฉพาะที่ ร่วมกับข้อดีของอัลกอริทึม Enhanced stochastic evolutionary (ESE) ที่นำเสนอโดย Jin, et al. (2005) ซึ่งได้อธิบายไปในบทที่ 2 ในส่วนของ

กระบวนการค้นหาเชิงกลุ่ม ที่สามารถปรับตัวเอง และเลือกกระบวนการค้นหาได้ และเพิ่มเติมในส่วนของการวิเคราะห์เงื่อนไขการหยุดทำงานที่เหมาะสมเข้าไป เพื่อให้การค้นหาแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์มีประสิทธิภาพมากขึ้น โดยโครงสร้างของอัลกอริทึม EESE ประกอบด้วยการทำงาน 2 ลูปซ้อนกัน เช่นเดียวกับ ESE การทำงานลูปในเป็นกระบวนการค้นหาเพื่อสร้างแผนการทดลองใหม่  $X_{try}$  ดังภาพ 12 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

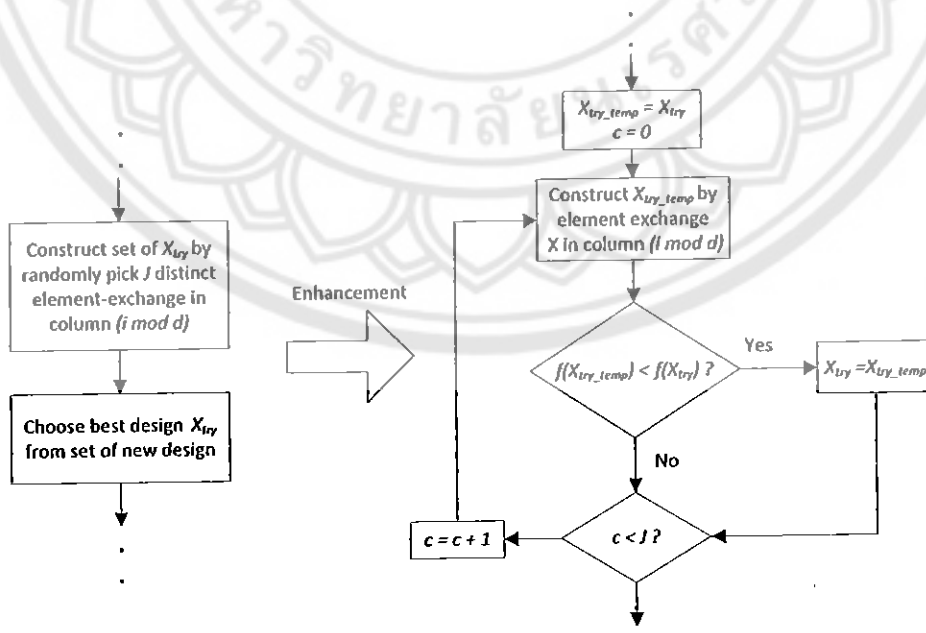
### กลไกของลูปใน (Inner loop)

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 ก่อนเข้าสู่การทำงานลูปในต้องมีการทำงานในส่วนของลูปนอกซึ่งครอบคลุมอยู่ คือ 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



การพัฒนาวิธีสร้างแผนการทดลองใหม่ในขั้นตอนที่ 2 ของลูปใน

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

การสุ่มครั้งที่ 1 ตรวจสอบกับข้อมูลใน อาร์เรย์ 0 ครั้ง

การสุ่มครั้งที่ 2 ตรวจสอบกับข้อมูลใน อาร์เรย์ 1 ครั้ง

.

.

.

.

.

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  $X_{try}$  มีความเคร่งครัดน้อยลงดังนี้

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

หรือ ยอมรับด้วยความน่าจะเป็นตามสมการ (13)

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

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

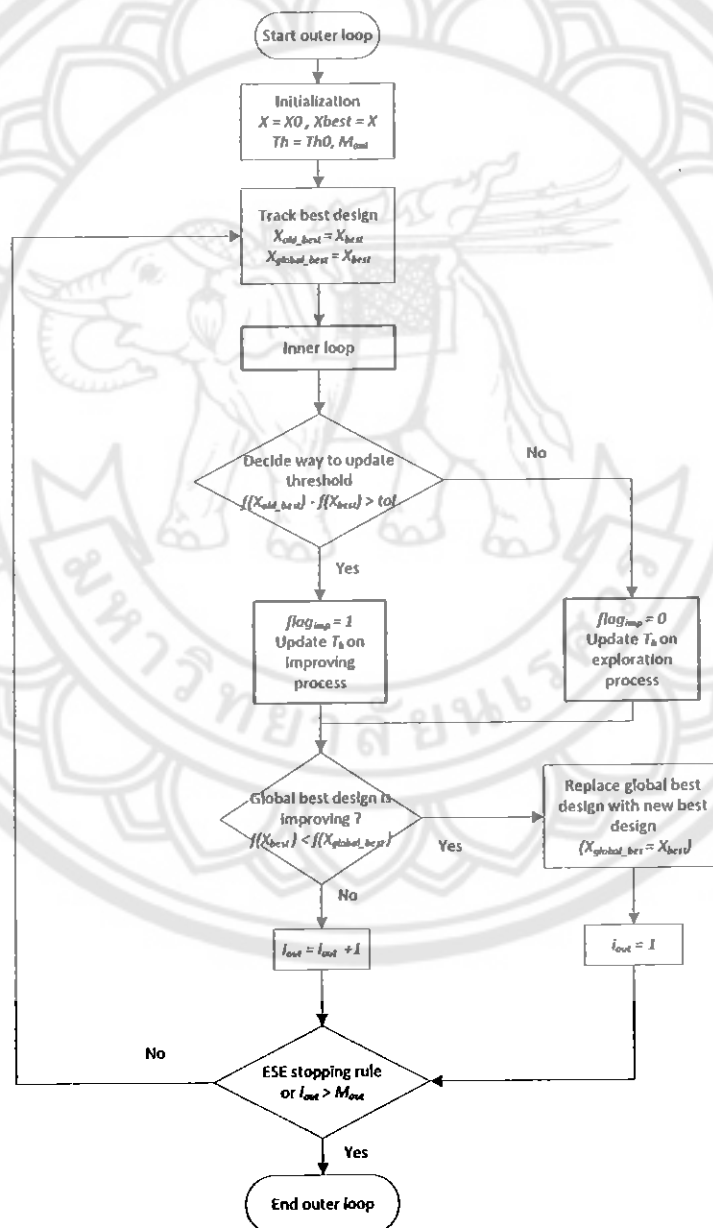
ระหว่าง (0, 1) และเพิ่มค่า  $n_{accept} = n_{accept} + 1$  เมื่อแผนการทดลอง  $X_{try}$  ถูกยอมรับ



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

ขั้นตอนที่ 5 การตรวจสอบเงื่อนไขการหยุดทำงาน

การพัฒนาวิธีการยอมรับแผนการทดลองในขั้นตอนที่ 3 นั้น สามารถช่วยลดจำนวนรอบการทำงานลูปในโดยต้องทำไปควบคู่กับการเพิ่มเงื่อนไขการหยุดเข้าไปในกฎการหยุดสืบค้นของลูปในขั้นตอนที่ 5 ซึ่งถ้าตรงตามเงื่อนไขใดเงื่อนไขหนึ่งให้หยุดการทำงาน 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 ทำให้สามารถแน่ใจได้ว่ารอบการทำงานของลูปในจะไม่เพิ่มขึ้นจากอัลกอริทึม ESE เดิมอย่างแน่นอน



Enhancement of enhanced stochastic evolutionary flowchart (ลูปนอก)

### กลไกของลูปนอก (Outer loop)

การทำงานของลูปนอกเป็นการนำเอาข้อดีของการค้นหาแบบเชิงกลุ่ม (Global search) ของอัลกอริทึม ESE มาใช้ ลูปนอกทำหน้าที่ควบคุมกระบวนการค้นหาโดยการปรับค่าพารามิเตอร์ควบคุม  $T_h$  ซึ่งในตอนเริ่มต้นการทำงานของอัลกอริทึมค่า  $T_h$  จะถูกตั้งให้ไว้ต่ำ (จะกล่าวถึงในส่วนของการขั้นตอนการทำงาน) 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

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

ขั้นตอนที่ 2 บันทึกแผนการทดลองที่ดีที่สุดก่อนเข้าสู่การทำงานของลูปใน  $X_{old\_best} = X_{best}$  เพื่อใช้เปรียบเทียบกับแผนการทดลองที่ดีที่สุดหลังจบการทำงานของลูปใน เพื่อให้ตรวจสอบเงื่อนไขการเลือกกระบวนการค้นหา 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  $X_{global\_best}$  หรือไม่ ตามสมการ  $\Phi_p(X_{best}) < \Phi_p(X_{global\_best})$  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  $n_{acpt}/M > 10\%$  และ อัตราส่วน  $n_{imp}/M < n_{acpt}/M$

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

3. นอกเหนือจากสองเงื่อนไขแรกจะทำการเพิ่ม  $T_h$  ตามสมการ  $T_h = T_h/\alpha_1$  โดยงานวิจัยนี้ กำหนดให้  $\alpha_1 = 0.8$  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.  $T_h$  เพิ่มตามสมการ  $T_h = T_h/\alpha_3$  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 EESE

ขั้นตอนนี้ผู้วิจัยได้ทำการพัฒนาโดยการเพิ่มเงื่อนไขการหยุดเข้าไปอีก 1 เงื่อนไขในกฎการหยุดสืบค้นของ ESE เดิมทำให้อัลกอริทึมไม่ต้องสร้างแผนการทดลองให้ครบตามจำนวนรอบการทำงานที่ตั้งไว้ถ้าค่า  $\Phi_p$  จากการทำงานของ EESE ไม่ดีขึ้นเป็นจำนวนหลายๆ รอบติดกัน

การเปลี่ยนวิธีการประเมินคุณสมบัติแผนการทดลอง (Re - calculate  $\Phi_p$ )

การแลกเปลี่ยนสมาชิกของแผนการทดลอง  $X$  เพื่อสร้างแผนการทดลอง  $X_{try}$  ด้วยวิธีที่กล่าวไปในตอนต้น สามารถใช้วิธีการประเมินค่า  $\Phi_p$  แบบใหม่ซึ่งมีผลทำให้สามารถช่วยลดระยะเวลาในการคำนวณ จากเดิมที่การประเมินค่า  $\Phi_p$  ต้องมีกระบวนการดังนี้

$\Phi_p$  เกณฑ์การเลือกค่าเหมาะสมที่นำเสนอโดย Morris and Mitchell (1995) ซึ่งปรับปรุงมาจาก Maximin distance สำหรับแผนการทดลอง  $X$  ระยะทางระหว่างจุดทดลองบนปริภูมิการทดลองแบบยูคลิด (Euclidean distance) สามารถหาโดยใช้สมการ

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

เมื่อคำนวณหาระยะทางระหว่างจุดทุกคู่ในแผนการทดลองซึ่งเขียนอยู่ในรูปของเมทริกซ์จัตุรัส

$D = [d_{ik}]_{m \times m}$  ให้ทำการเรียงระยะทางระหว่างจุดทั้งหมดจากน้อยไปมาก ( $d_1, d_2, \dots, d_m$ )

และ สร้างดัชนี (Index) ( $j_1, j_2, \dots, j_m, j_j$ ) คือ จำนวนของคู่จุดที่แบ่งตามระยะทาง  $d_j$  เกณฑ์การเลือกค่าเหมาะสม  $\Phi_p$  สามารถคำนวณได้จากสมการ

$$\Phi_p = \left[ \sum_{j=1}^m j_j d_j^{-p} \right]^{\frac{1}{p}}$$

วิธีการคำนวณให้ง่ายขึ้นโดยไม่นำค่าของ  $j_j$  มาเกี่ยวข้องตามสมการ (18) ซึ่งโดยทั่วไปแล้วแผนการทดลองแบบ LHD ที่เหมาะสมคือแผนการทดลองที่ให้ค่า  $\Phi_p$  น้อยที่สุด

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

ซึ่งจะใช้วิธีการประเมินค่า  $\Phi_p$  ตามสมการ (18) กับแผนการทดลอง  $X_0$  เท่านั้น แผนการทดลองที่สร้างขึ้นหลังจากนี้ ใช้วิธีการดังต่อไปนี้

การประเมินค่า  $\Phi_p$  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 สำหรับทุก ๆ  $1 < j < n$  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  $d'_{i_2 j}$  คือระยะทางระหว่างจุดใหม่ที่จะถูกนำไปแทนที่ระยะทางระหว่างจุดเดิมใน  $D$  แล้วจึงคำนวณหาค่า  $\Phi_p$  แบบใหม่ โดยใช้ระยะทางระหว่างจุดที่เปลี่ยนแปลงไปหลังการแลกเปลี่ยนสมาชิกกับค่า  $\Phi_p$  เดิมก่อนการแลกเปลี่ยนสมาชิกดังสมการ

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

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

ผลการพิสูจน์การลดระยะเวลาในการคำนวณคือการคำนวณหาค่า  $\Phi_p$  แบบเก่าตามสมการ (18) ประกอบด้วย 2 ส่วนย่อย คือ การหาค่าระยะห่างจุดทดลองบนปริภูมิการทดลองแบบยูคลิด (Euclidean distance) ในสมการ (16) มีความซับซ้อนในการคำนวณเท่ากับ  $O(dn^2)$ , และการคำนวณหาค่า  $\Phi_p$  โดยไม่นำค่าของ  $j$  มาเกี่ยวข้องตามสมการ (18) มีความซับซ้อนในการคำนวณเท่ากับ  $O(n^2 \log_2 p)$  ดังนั้นเวลาที่ใช้ทั้งหมดเท่ากับ  $O(dn^2 + n^2 \log_2 p)$  ซึ่งจะใช้เวลาานมากเมื่อตัวแปรเข้าหรือ  $n$  มีขนาดใหญ่ขึ้น (Jin, Chen and Sudjianto, 2005) ส่วนการคำนวณค่า  $\Phi_p$  แบบเก่าตามสมการ (22) ประกอบด้วย 2 ส่วนย่อย คือ การระยะทางระหว่างจุดใหม่ที่จะถูกนำไปแทนที่ระยะทางระหว่างจุดเดิมใน  $D$  ตามสมการ (19) - (21) มีความซับซ้อนในการคำนวณเท่ากับ  $O(n)$  และการคำนวณค่า  $\Phi_p$  แบบใหม่ตามสมการ (22) คำนวณเท่ากับ  $O(n^2 \log_2 p)$  ดังนั้นเวลาที่ใช้ทั้งหมดเท่ากับ  $O(n + n^2 \log_2 p)$  ซึ่งจะเห็นได้ว่าวิธีการคำนวณค่า  $\Phi_p$  แบบใหม่มีความซับซ้อนในการคำนวณที่น้อยกว่าแบบเก่า ซึ่งมีผลให้ใช้เวลาการคำนวณลดลง



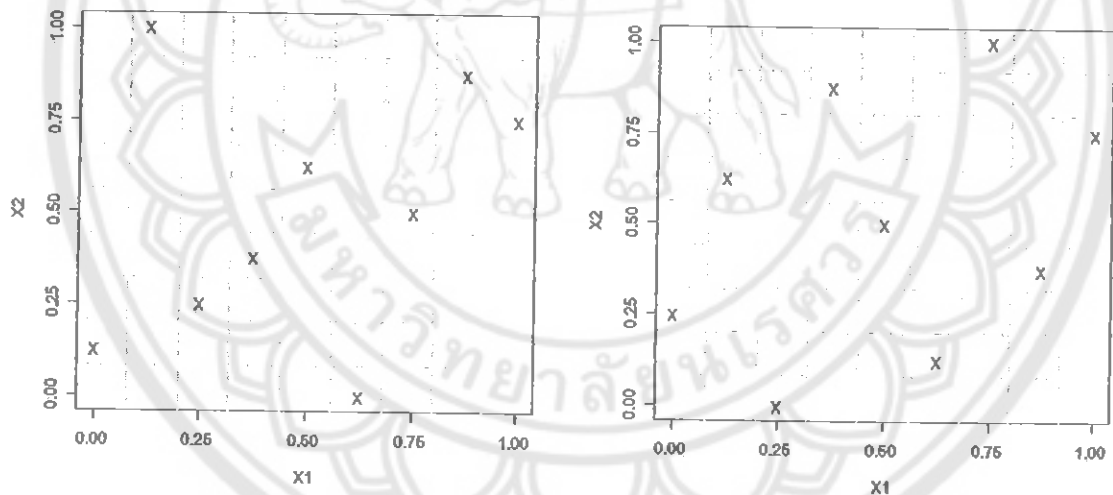
## บทที่ 4

### ผลการวิจัย

บทนี้ จะกล่าวถึงแผนการทดลองที่เหมาะสมที่สร้างจากอัลกอริธึมการสืบค้นที่น่าเสนอมาทั้งหมด โดยจะแสดงการเปรียบเทียบประสิทธิภาพของอัลกอริธึมการสืบค้น โดยแบ่งการเปรียบเทียบเป็นสองส่วนคือ เปรียบเทียบในส่วนของคุณสมบัติของแผนการทดลอง (Accuracy) คือการพิจารณาค่า  $\Phi_p$  ของแผนการทดลองที่เหมาะสมเมื่อสิ้นสุดการทำงานของอัลกอริธึม และเปรียบเทียบในส่วนของเวลาที่ใช้ในการทำงานของอัลกอริธึม (Efficiency) ภายใต้ขนาดของแผนการทดลองที่แตกต่างกัน

#### แผนการทดลองที่เหมาะสม (Optimal design)

อัลกอริธึมการสืบค้นทำหน้าที่สร้างแผนการทดลองที่เหมาะสม ซึ่งในงานวิจัยนี้ได้เลือกใช้กลุ่มการออกแบบการทดลอง LHD



(a) Before optimization

(b) After optimization

แผนการทดลอง LHD ขนาด  $9 \times 2$  ก่อนและหลังการใช้อัลกอริธึมการสืบค้นแบบสุ่มเชิงวิวัฒนาการ

ภาพ (a) แสดงตัวอย่างแผนการทดลองแบบ LHD ก่อนการใช้อัลกอริธึมการสืบค้น จะเห็นว่าจุดของการทดลองมีการครอบคลุมพื้นที่ในปริภูมิการทดลองน้อย หรือมีคุณสมบัติแบบการเติมเต็มปริภูมิ (Space filling) ต่ำ เมื่อแผนการทดลองนี้ถูกนำไปประเมินหาค่า  $\Phi_p$  จะทำให้ได้ค่า  $\Phi_p = 6.764$  ซึ่งถือว่าสูงเมื่อเปรียบเทียบกับแผนการทดลองภาพ (b) ที่มีคุณสมบัติการเติมเต็มปริภูมิที่สูง ซึ่งให้ค่า  $\Phi_p = 4.273$  ที่ต่ำกว่า โดยแผนการทดลองในภาพ (b) สร้างจากอัลกอริธึมการสืบค้นแบบสโตแคสติกเชิงวิวัฒนาการ มีผลทำให้แผนการทดลองมีคุณสมบัติเติมเต็มปริภูมิที่สูงขึ้น ส่งผลให้เมื่อนำไปสร้าง

โมเดลการพยากรณ์ จะทำให้ได้โมเดลที่มีความแม่นยำสูงในการทำนาย (Prediction) ประสิทธิภาพหรือผลผลิตของระบบ โดยในการศึกษาครั้งนี้มุ่งเน้นทดสอบบนแผนการทดลองขนาดต่างๆ ที่จะกล่าวถึงในส่วนต่อไป

**คุณภาพของข้อมูล (Data quality)**

ขนาดของแผนการทดลองสำหรับการศึกษาในครั้งนี้จะกำหนดโดยใช้หลักการประมาณค่าพหามิเตอร์ที่เป็นไปได้ทั้งหมดตามรูปแบบของสมการพหุนามดีกรีสอง (Second order polynomial) (Montgomery, 2005) ดังสมการ

$$n = 2d + 4 \binom{d}{2} + 1$$

เมื่อ  $n$  = จำนวนการรัน และ  $d$  = จำนวนตัวแปรเข้า  
เมื่อจำนวนตัวแปรที่ใช้มีขนาดตั้งแต่ 2 - 20 จะส่งผลให้จำนวนการรันมีขนาดดังตาราง

แสดงขนาดแผนการทดลองแบบ LHD ที่ใช้ในการทดสอบ

$d$	2	3	4	5	6	7	8	9	10	11	12	13	14	15	20
$n$	9	19	33	51	73	99	129	163	201	243	289	339	393	451	801

การศึกษานี้จะให้อัลกอริทึมทำงาน (Execute) ซ้ำ 10 ครั้ง (โดยใช้ตัวแปลภาษา R บนเครื่องคอมพิวเตอร์ที่ใช้หน่วยประมวลผล AMD Athlon 64 X2 3.0 GHz.) จากแผนการทดลองเริ่มต้นที่แตกต่างกันจากการสุ่ม แล้ววิเคราะห์ผลการทำงานของอัลกอริทึมในเชิงเปรียบเทียบเมื่อสิ้นสุดการ และใช้วิธีการทางสถิติในการวิเคราะห์ข้อมูลคือ ค่าเฉลี่ย (Mean), ค่าเบี่ยงเบนมาตรฐาน (S.D.) ค่าสูงสุด (Max) และค่าต่ำสุด (Min)

**ผลการทดสอบและเปรียบเทียบ**

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

เปรียบเทียบคุณสมบัติของแผนการทดลองคือการพิจารณาค่า  $\Phi_p$  ของแผนการทดลองที่เหมาะสมเมื่อสิ้นสุดการทำงานของอัลกอริทึม และใช้ค่าทางสถิติในการวิเคราะห์ข้อมูลคือค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน ค่าสูงสุด และค่าต่ำสุด และเปรียบเทียบการลดลงของค่า  $\Phi_p$  เมื่ออัลกอริทึมมีจำนวนรอบการแลกเปลี่ยนสมาชิกที่เท่ากัน โดยกระทำกับแผนการทดลองขนาด 9 x 2 51 x 5, 201 x 10 และ 801 x 20

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

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

#### 1. ผลการทดสอบกับปัญหาขนาดเล็ก

แผนการทดลองที่มีตัวแปรเข้าจำนวน 2 - 9 ตัวแปร การตั้งค่าพารามิเตอร์สำหรับแต่ละอัลกอริทึมมีรายละเอียดดังนี้ MSA มีจำนวนรอบการทำงาน  $i_m = 1000$  สำหรับแผนการทดลองขนาด 2 - 6 ตัวแปร และ  $i_m = 500$  สำหรับแผนการทดลองขนาด 7 - 9 ตัวแปร (Thamma, et al. 2008) ส่วน ESE และ EESE ตั้งค่าให้มีจำนวนรอบการทำงานรูปนอกคือ 40 สำหรับแผนการทดลองขนาด 2 - 6 ตัวแปร และ 20 สำหรับแผนการทดลองขนาด 7 - 9 ตัวแปร สำหรับผลการทดสอบกับแผนการทดลองขนาดเล็ก เปรียบเสมือนการทดสอบวิธีการค้นหาแบบเชิงกลุ่มในกระบวนการค้นหาแบบสำรวจ (Exploration process) ของ ESE และ EESE เนื่องจากแผนการทดลองขนาดเล็ก การเพิ่มจำนวนรอบการทำงานรูปนอกให้กับ ESE และ EESE เป็นการเพิ่มเพื่อให้กระบวนการของ ESE และ EESE ให้สามารถหลุดออกจากการค้นหาจากแผนการทดลองที่เหมาะสมแบบเฉพาะที่จึงมีผลทำให้ค่า  $\Phi_p$  ลดลงเพียงเล็กน้อยเท่านั้น

#### เปรียบเทียบคุณสมบัติของแผนการทดลอง

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



สรุปเมื่อทำการพิจารณาโดยภาพรวมแล้วพบว่า ESE สามารถสร้างแผนการทดลองที่มีคุณสมบัติที่ดีกว่า MSA และ EESE คือให้ค่า  $\Phi_p$  โดยเฉลี่ยที่ต่ำกว่าทั้ง MSA และ EESE ที่สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  ใกล้เคียงกัน และเมื่อพิจารณาค่าส่วนเบี่ยงเบนมาตรฐาน โดยภาพรวมแล้วพบว่าทั้ง MSA, ESE และ EESE มีการกระจายน้อย โดยเฉพาะเมื่อแผนการทดลองมีขนาดใหญ่ขึ้น ซึ่งแสดงให้เห็นถึงการทำงานอย่างคงที่ (Consistency) ในแต่ละอัลกอริทึม แสดง  $\Phi_p$  ในเชิงสถิติที่ได้จาก MSA, ESE และ EESE ของปัญหาขนาดเล็ก

$(p = 5, l = 2)$					
LHDs	Algorithm	Min	Max	Mean	SD
9 × 2	MSA	4.273	4.273	4.273	0
	ESE	4.273	4.344	4.287	0.029
	EESE	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
	EESE	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
	EESE	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
	EESE	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
	EESE	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
	EESE	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
	EESE	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
	EESE	6.039	6.044	6.042	0.001

แสดงประสิทธิภาพของ MSA, ESE และ EESE ของปัญหาขนาดเล็ก

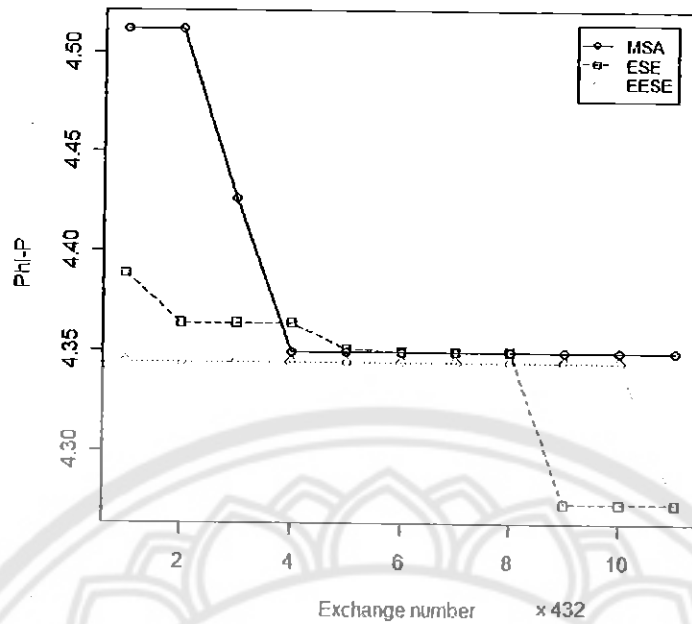
LHDs	Algorithm	Performance (Average)		Time ratio		
		Time (sec.)	#Exchange	MSA/ESE	MSA/EESE	ESE/EESE
2 × 9	MSA	19.993	47140			
	ESE	3.595	5760	5.561	8.707	1.565
	EESE	2.296	5415			
19 × 3	MSA	123.441	133418			
	ESE	67.74	41040	1.822	3.601	1.976
	EESE	34.275	37121			
33 × 4	MSA	351.293	212889			
	ESE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

### เปรียบเทียบประสิทธิภาพของอัลกอริธึม

ตารางเสนอผลการเปรียบเทียบประสิทธิภาพของ MSA, ESE และ EESE โดยพิจารณา เวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสม (Time) ในหน่วยวินาที (Second), จำนวนครั้งของการแลกเปลี่ยนสมาชิก (#Exchange) และอัตราส่วนเวลาการสร้างแผนการทดลองที่เหมาะสมระหว่าง MSA/ESE, MSA/EESE และ ESE/EESE ซึ่งผลที่ได้จากตารางแสดงให้เห็นว่า สำหรับแผนการทดลองขนาด  $9 \times 2$  อัลกอริธึม EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมโดยเฉลี่ยน้อยที่สุดคือ 2.296 วินาที โดยใช้เวลาน้อยกว่า ESE ที่ตามมาเป็นอันดับสองคือ 3.595 วินาที อยู่ประมาณ 1.565 เท่า และน้อยกว่า MSA ที่ใช้เวลา 19.993 วินาที และเมื่อพิจารณาที่จำนวนครั้งการแลกเปลี่ยนสมาชิก EESE ใช้จำนวนครั้งการแลกเปลี่ยนสมาชิกที่น้อยกว่า ESE อยู่เล็กน้อยคือ 5,415 และ 5,760 ตามลำดับซึ่งน้อยกว่า MSA ที่ใช้จำนวนครั้งการแลกเปลี่ยนสมาชิกไปถึง 47,140 ครั้งในการสร้างแผนการทดลองที่เหมาะสม และสำหรับแผนการทดลองขนาด  $51 \times 5$  อัลกอริธึม EESE ก็ยังสามารถสร้างแผนการทดลองที่เหมาะสมโดยใช้เวลาและจำนวนครั้งการแลกเปลี่ยนสมาชิกโดยเฉลี่ยน้อยที่สุดคือ 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

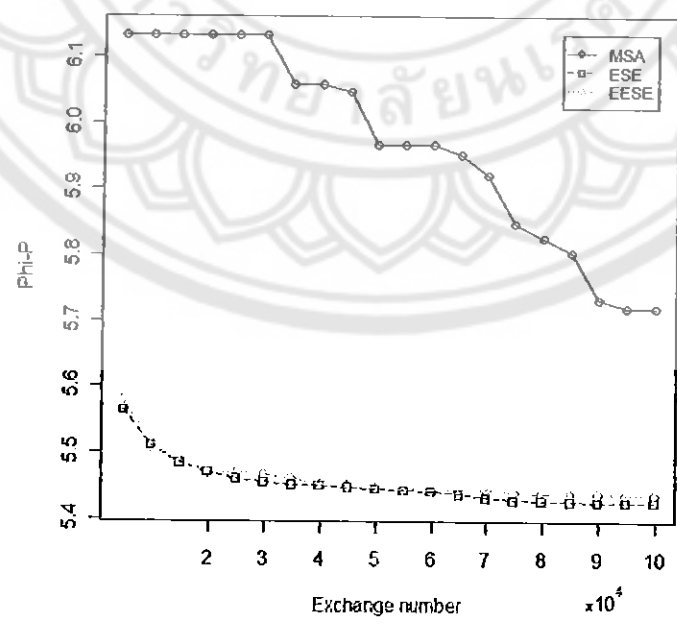
สรุปสำหรับทุกขนาดของของแผนการทดลองในปัญหาขนาดเล็ก อัลกอริธึม ESE และ EESE ใช้จำนวนครั้งของการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองน้อยกว่า MSA เกิน 50% เกือบทุกขนาดของปัญหา ซึ่งส่งผลให้ใช้นิเวศน้อยกว่าด้วย ซึ่งเห็นว่า ESE และ EESE มีประสิทธิภาพเหนือกว่า MSA ในภาพรวม และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่า EESE ใช้จำนวนครั้งในการแลกเปลี่ยนสมาชิกน้อยกว่า ESE ส่งผลให้ใช้นิเวศน้อยกว่าโดยเฉลี่ยประมาณ 1.65 เท่าในทุกขนาดของแผนการทดลอง

การพิจารณาเปรียบเทียบการลดลงของค่า  $\Phi_p$  ของแผนการทดลองขนาด  $9 \times 2$  โดยให้แต่ละอัลกอริธึมทำการแลกเปลี่ยนสมาชิกเพื่อสร้างแผนการทดลองใหม่จำนวน 4,320 รอบ จากแผนการทดลองเริ่มต้นที่แตกต่างกันโดยการสุ่ม กระทำทั้งหมด 10 ครั้งแล้วทำการหาค่าเฉลี่ยของจำนวนการครั้งแลกเปลี่ยนสมาชิก เพื่อพิจารณาการลดลงของค่า  $\Phi_p$  ซึ่งเป็นการแสดงให้เห็นว่าแต่ละอัลกอริธึมสามารถสร้างแผนการทดลองที่เข้าใกล้แผนการทดลองที่เหมาะสม



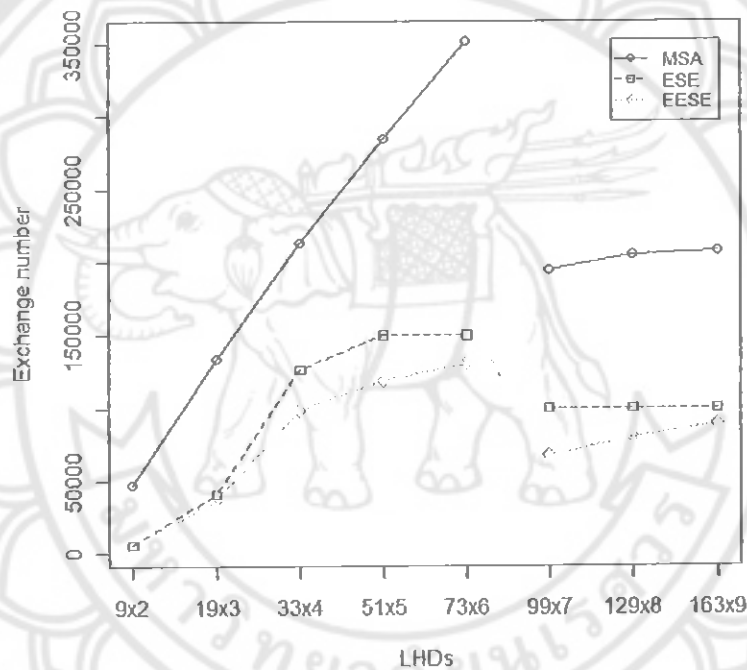
**การลดลงของค่า  $\Phi_p$  ของแผนการทดลองขนาด 9 x 2 ต่อจำนวนรอบการแลกเปลี่ยนสมาชิก**

จากภาพแสดงกราฟแสดงการลดลงของค่า  $\Phi_p$  โดยเริ่มต้นแสดงเมื่อแต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปจำนวน 432 รอบ ผลที่ได้พบว่าค่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้นโดย ESE ( $\Phi_p = 4.273$ ) และ EESE ( $\Phi_p = 4.273$ ) ลดลงเร็วกว่า MSA ( $\Phi_p = 4.349$ ) เมื่อแต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปประมาณ 4,320 รอบ มีผลทำให้ ESE และ EESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยกว่า MSA และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้นโดย ESE ลดลงเร็วกว่า EESE 864 รอบของการแลกเปลี่ยนสมาชิก



**การลดลงของค่า  $\Phi_p$  ของแผนการทดลองขนาด 51 x 5 ต่อจำนวนรอบการแลกเปลี่ยนสมาชิก**

การเปรียบเทียบการลดลงของค่า  $\Phi_p$  ของแผนการทดลองขนาด  $51 \times 5$  โดยให้แต่ละอัลกอริทึมทำการแลกเปลี่ยนสมาชิกเพื่อสร้างแผนการทดลองใหม่จำนวน 100,000 รอบ จากแผนการทดลองเริ่มต้นที่แตกต่างกันโดยการสุ่ม กระทำทั้งหมด 10 ครั้งแล้วทำการหาค่าเฉลี่ยของจำนวนการครั้งแลกเปลี่ยนสมาชิก เพื่อพิจารณาการลดลงของค่า  $\Phi_p$  ซึ่งเป็นการแสดงให้เห็นว่าแต่ละอัลกอริทึมสามารถสร้างแผนการทดลองที่เข้าใกล้แผนการทดลองที่เหมาะสม ผลที่ได้คือการลดลงของค่า  $\Phi_p$  ในแผนการทดลองขนาด  $51 \times 5$  โดยเริ่มต้นแสดงเมื่อแต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปจำนวน 5,000 รอบ ผลที่ได้แสดงให้เห็นอย่างชัดเจนว่าค่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้นจาก ESE ( $\Phi_p = 5.424$ ) และ EESE ( $\Phi_p = 5.436$ ) ลดลงเร็วกว่า MSA ( $\Phi_p = 5.719$ ) เมื่อแต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไป 100,000 รอบ เมื่อแผนการทดลองมีขนาดเพิ่มขึ้นจะเริ่มเห็นความแตกต่างของค่า  $\Phi_p$  และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้นโดย ESE ลดลงเร็วกว่า EESE เล็กน้อย และเมื่อจำนวนรอบการแลกเปลี่ยนสมาชิกเพิ่มมากขึ้น



### จำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมกับปัญหาขนาดใหญ่

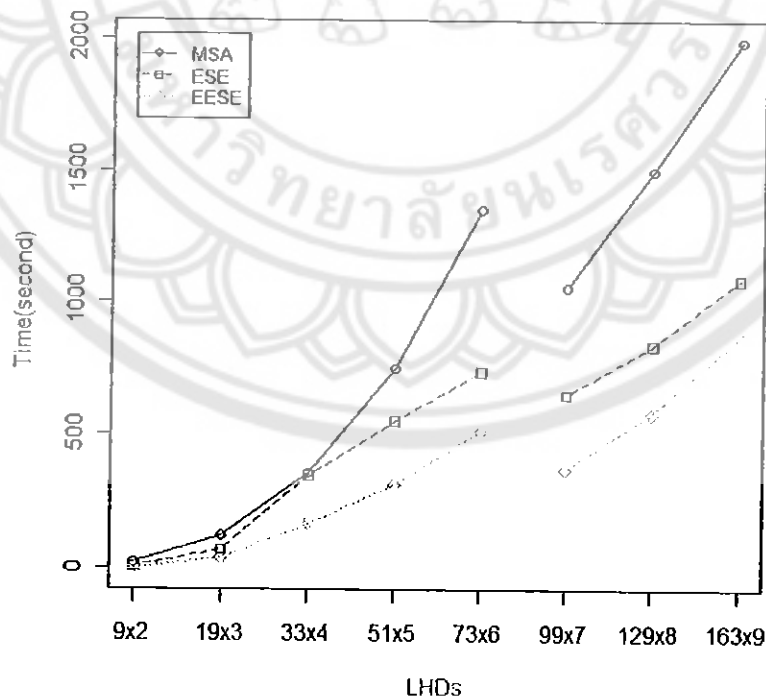
แสดงการเปรียบเทียบจำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมปัญหาขนาดเล็ก โดยการพิจารณาที่ค่า  $\Phi_p$  ที่ต่ำที่สุดของแต่ละอัลกอริทึมหลังจบการทำงานจำนวน 10 ครั้ง แล้วจึงนำไปหาค่าเฉลี่ย ผลที่ได้คือ สำหรับแผนการทดลองขนาด  $9 \times 2$  MSA ใช้จำนวนครั้งการแลกเปลี่ยนสมาชิกสูงสุด (47,140 รอบ) ถัดมาคือ ESE (5,760 รอบ) และ EESE (5,415 รอบ) ตามลำดับ ในการสร้างแผนการทดลองที่ให้ค่า  $\Phi_p = 4.273$  และสำหรับแผนการทดลองขนาด  $73 \times 6$  MSA ใช้จำนวนครั้งการแลกเปลี่ยนสมาชิกสูงถึง 351,624 รอบ ซึ่งสูงกว่า ESE (150,000 รอบ) ประมาณ 2.4 เท่า และ EESE (130,500 รอบ) ประมาณ 2.5 เท่า ในการสร้างแผนการทดลองที่ให้ค่า  $\Phi_p = 5.590$  สรุปโดยรวมคือ MSA ใช้จำนวนรอบสูงสุดในการสร้างแผนการทดลองที่เหมาะสม รองลงมาคือ ESE และ EESE ตามลำดับ แต่เมื่อขนาดของแผนการทดลองเพิ่มขึ้นจำนวนรอบการ

แลกเปลี่ยนสมาชิกที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมสำหรับ EESE จะเริ่มใกล้เคียงกับ ESE มากขึ้นเรื่อย โดยสังเกตได้จากแผนการทดลองขนาด  $163 \times 9$  (EESE = 88,955 รอบ และ ESE = 100,000) สาเหตุเนื่องมาจากกฎการหยุดสับสน (ที่กล่าวไปในบทที่ 3) โดยจากการศึกษาพบว่าเมื่อปัญหามีขนาดเพิ่มขึ้น แผนการทดลองสุดท้ายที่ได้ในแต่ละรอบการทำงานของอัลกอริทึมจะให้ค่า  $\Phi_p$  ที่น้อยลงเสมอในทุกๆ รอบ ซึ่งเป็นเงื่อนไขที่ไม่สอดคล้องกับกฎการหยุดสับสนที่ผู้วิจัยเลือกใช้

การพิจารณาเปรียบเทียบเวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมกับปัญหาขนาดเล็ก โดยการพิจารณาที่ค่า  $\Phi_p$  ที่ต่ำที่สุดของแต่ละอัลกอริทึมหลังจบการทำงานจำนวน 10 ครั้ง แล้วจึงนำเวลาที่ได้ไปหาค่าเฉลี่ย

ภาพ 19 กราฟแสดงเวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมของ MSA, ESE และ EESE สำหรับแผนการทดลองขนาด  $9 \times 2$  เวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมในหน่วยวินาทีของ MSA, ESE และ EESE นั้นใกล้เคียงกัน (MSA = 19.993, ESE = 3.595 และ EESE = 2.296) และจะเริ่มเห็นความแตกต่างของเวลาที่ใช้เมื่อแผนการทดลองมีขนาดใหญ่ขึ้น โดยสามารถเห็นได้อย่างชัดเจนในแผนการทดลองขนาด  $73 \times 6$  ว่า MSA ใช้เวลามากที่สุดในการสร้างแผนการทดลองที่เหมาะสมคือ 1352.469 วินาที มากกว่า ESE ที่ใช้เวลา 734.981 วินาที ถึง 1.84 เท่า และมากกว่า EESE ที่ใช้เวลา 512.225 ถึง 2.64 เท่า

สรุป โดยรวมคือ MSA จะใช้เวลานานกว่า ESE และ EESE ในการสร้างแผนการทดลองที่เหมาะสมเพิ่มมากขึ้นเรื่อยๆ เมื่อแผนการทดลองมีขนาดใหญ่ขึ้น ในขณะที่ EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมน้อยกว่า ESE เฉลี่ยประมาณ 1.66 เท่าในทุกขนาดของแผนการทดลองในปัญหาขนาดเล็ก



เวลาในการสร้างแผนการทดลองที่เหมาะสมกับปัญหาขนาดเล็ก

จากการทดสอบอัลกอริทึมการสืบค้นกับปัญหาขนาดเล็ก สามารถสรุปได้ว่า MSA เป็นอัลกอริทึมที่ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมกว่าอัลกอริทึม ESE และ EESE โดยสามารถสร้างแผนการทดลองที่มีคุณสมบัติที่ใกล้เคียงกับ EESE ( $\Phi_p$  ใกล้เคียงกัน) ในขณะที่ ESE ให้แผนการทดลองที่มีคุณสมบัติที่ดีกว่า ซึ่งให้ค่า  $\Phi_p$  ที่ต่ำกว่า และต่ำที่สุดในเกือบทุกขนาดของแผนการทดลองในปัญหาขนาดเล็ก แต่ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมเป็นอันดับสองรองจาก EESE ในขณะที่ EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมน้อยที่สุด และคุณสมบัติของแผนการทดลองอยู่ในเกณฑ์ที่สามารถยอมรับได้

## 2. ผลการทดสอบกับปัญหาขนาดใหญ่

แผนการทดลองที่มีตัวแปรเข้าจำนวน 10 - 15 และ 20 ตัวแปร ซึ่งการตั้งค่าพารามิเตอร์สำหรับแต่ละอัลกอริทึมมีรายละเอียดดังนี้ MSA มีจำนวนรอบการทำงาน  $n_m = 500$  ส่วน ESE และ EESE ตั้งค่าจำนวนรอบการทำงานลูปนอกคือ 20 สำหรับปัญหาขนาดใหญ่การสร้างแผนการทดลองที่เหมาะสมของ MSA, ESE และ EESE จะใช้เวลานานมากขึ้น และคุณสมบัติของแผนการทดลองที่พิจารณาจากค่า  $\Phi_p$  มีความแตกต่างกันอย่างชัดเจน ซึ่งมีรายละเอียดของผลการทดสอบดังนี้

### เปรียบเทียบคุณสมบัติของแผนการทดลอง

ตาราง 5 แสดงค่าสถิติพื้นฐานเช่น ค่าเฉลี่ย (Mean) ค่าต่ำสุด (Min) ค่าสูงสุด (Max) และค่าส่วนเบี่ยงเบนมาตรฐาน (S.D.) ของค่า  $\Phi_p$  จากการทำงานซ้ำของแต่ละอัลกอริทึมจำนวน 10 ครั้ง ซึ่งผลที่ได้คือสำหรับแผนการทดลองขนาด  $201 \times 10$  อัลกอริทึม ESE สามารถสร้างแผนการทดลองที่มีคุณสมบัติที่ดีที่สุดซึ่งให้ค่า  $\Phi_p$  โดยเฉลี่ยเท่ากับ 6.172 และมีค่าส่วนเบี่ยงเบนมาตรฐานคือ 0.001 เท่านั้น รองลงมาคือ EESE ที่สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  โดยเฉลี่ยเท่ากับ 6.174 และมีค่าส่วนเบี่ยงเบนมาตรฐานเป็น 0.000 โดยแผนการทดลองที่สร้างจาก MSA ให้ค่า  $\Phi_p$  โดยเฉลี่ยสูงถึง 6.180 และสำหรับแผนการทดลองขนาด  $801 \times 20$  อัลกอริทึม ESE และ EESE สร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  โดยเฉลี่ยต่ำเท่ากันคือ 7.254 และแผนการทดลองที่สร้างจาก MSA ให้ค่า  $\Phi_p$  โดยเฉลี่ยเท่ากับ 7.272

สรุปคือ MSA ไม่สามารถสร้างแผนการทดลองที่เหมาะสมที่ให้ค่า  $\Phi_p$  ต่ำกว่า ESE และ EESE เมื่อจบการทำงานของอัลกอริทึม ซึ่งบ่งชี้ให้เห็นว่ากระบวนการค้นหาของ MSA ไม่เหมาะสมกับการสร้างแผนการทดลองที่มีจำนวนตัวแปรเข้าจำนวนมากกว่า 10 ตัวแปร และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่าสามารถสร้างแผนการทดลองที่เหมาะสม ที่ให้ค่า  $\Phi_p$  โดยเฉลี่ยเท่ากันในทุกขนาดของแผนการทดลอง สำหรับการทดสอบกับปัญหาขนาดใหญ่นั้นการเพิ่มจำนวนรอบการทำงานให้กับ ESE และ EESE ส่งผลให้กระบวนการค้นหาแบบปรับปรุง (Improving process) ของ ESE และ EESE สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  ที่ต่ำลงทำให้แผนการทดลองที่ได้มีคุณสมบัติดีขึ้น

แสดง  $\phi_p$  ในเชิงสถิติที่ได้จาก MSA, ESE และ EESE ของปัญหาขนาดใหญ่

LHDs	Algorithm	$(p = 5, t = 2)$			
		Min	Max	Mean	SD
201 × 10	MSA	6.179	6.181	6.180	0
	ESE	6.170	6.174	6.172	0.001
	EESE	6.173	6.175	6.174	0
243 × 11	MSA	6.308	6.312	6.310	0.001
	ESE	6.298	6.301	6.300	0
	EESE	6.299	6.301	6.300	0
289 × 12	MSA	6.432	6.435	6.434	0
	ESE	6.422	6.423	6.422	0
	EESE	6.422	6.423	6.422	0
339 × 13	MSA	6.552	6.554	6.553	0
	ESE	6.539	6.540	6.539	0
	EESE	6.539	6.540	6.539	0
393 × 14	MSA	6.666	6.668	6.667	0
	ESE	6.652	6.653	6.652	0
	EESE	6.652	6.653	6.652	0
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



แสดงประสิทธิภาพของ MSA, ESE และ EESE ของปัญหาขนาดใหญ่

LHDs	Algorithm	Performance (Average)		Time ratio		
		Time (sec.)	#Exchange	MSA/ESE	MSA/EESE	ESE/EESE
201 × 10	MSA	2795.741	219912			
	ESE	1349.788	100000	2.071	2.376	1.147
	EESE	<b>1176.529</b>	<b>94070</b>			
243 × 11	MSA	3809.109	226567			
	ESE	1655.321	100000	2.301	2.410	1.047
	EESE	<b>1579.913</b>	<b>99685</b>			
289 × 12	MSA	4813.372	232143			
	ESE	2049.986	100000	2.348	2.440	1.039
	EESE	<b>1972.439</b>	<b>99865</b>			
339 × 13	MSA	5885.643	229399			
	ESE	2540.851	100000	2.316	2.396	1.034
	EESE	<b>2455.954</b>	<b>100000</b>			
393 × 14	MSA	7268.430	234783			
	ESE	3060.171	100000	2.375	2.465	1.038
	EESE	<b>2947.597</b>	<b>100000</b>			
451 × 15	MSA	8686.660	234517			
	ESE	3648.646	100000	2.380	2.463	1.034
	EESE	<b>3526.691</b>	<b>100000</b>			
801 × 20	MSA	20854.01	260539			
	ESE	7744.424	100000	2.692	2.718	1.009
	EESE	<b>7672.487</b>	<b>100000</b>			

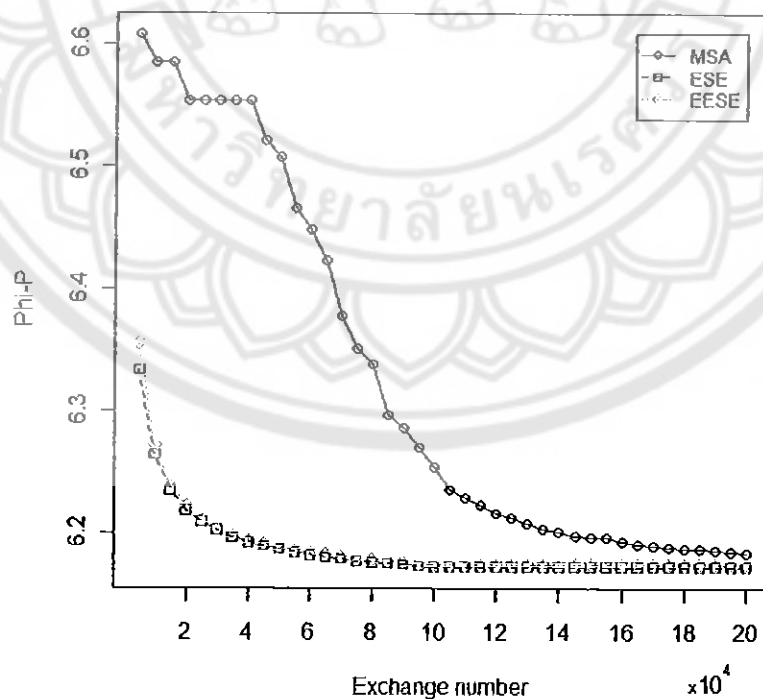
เปรียบเทียบประสิทธิภาพของอัลกอริธึม

ตารางเสนอผลการเปรียบเทียบประสิทธิภาพของ MSA, ESE และ EESE โดยพิจารณา เวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสม (Time) ในหน่วยวินาที (Second) จำนวนครั้งของการแลกเปลี่ยนสมาชิก (#Exchange) และอัตราส่วนเวลาการสร้างแผนการทดลองที่เหมาะสมระหว่าง

MSA/ESE, MSA/EESE และ ESE/EESE ซึ่งผลที่ได้สำหรับแผนการทดลองขนาด  $201 \times 10$  เวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมของ MSA โดยเฉลี่ยเท่ากับ 2795.741 วินาที ซึ่งมากกว่า ESE ที่ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมโดยเฉลี่ยเท่ากับ 1349.788 วินาที ถึง 2 เท่า โดย EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมโดยเฉลี่ยน้อยที่สุดคือ 1176.529 วินาที และเมื่อพิจารณาที่จำนวนรอบการแลกเปลี่ยนสมาชิก ESE ใช้จำนวนรอบการแลกเปลี่ยนสมาชิกมากกว่า EESE อยู่เล็กน้อย (ESE = 100,000 และ EESE = 94,070) โดยที่ MSA ใช้จำนวนรอบการแลกเปลี่ยนสมาชิกสูงที่สุดคือ 219,912 และสำหรับแผนการทดลองขนาด  $801 \times 20$  ซึ่งเป็นแผนการทดลองที่มีขนาดใหญ่ที่สุด เมื่อพิจารณาพบว่า ESE ใช้จำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมเท่ากับ EESE คือ 100,000 แต่ EESE ใช้เวลาเท่ากับ 7672.487 วินาที ซึ่งน้อยกว่า ESE ที่ใช้เวลา 7744.424 วินาทีอยู่น้อย

สรุปคือ MSA ใช้เวลา และจำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมสูงกว่า ESE และ EESE ในทุกขนาดของแผนการทดลอง และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่า EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมน้อยกว่า ESE ทุกขนาดของแผนการทดลอง แม้ว่าจำนวนรอบการแลกเปลี่ยนสมาชิกจะเท่ากัน สืบเนื่องมาจากการพัฒนาวิธีการสร้างแผนการทดลองใหม่ (ภาพ 13) ซึ่งมีผลทำให้เวลาในการสร้างแผนการทดลองต่อหนึ่งรอบการทำงานของ EESE ลดลงเมื่อเทียบกับ ESE

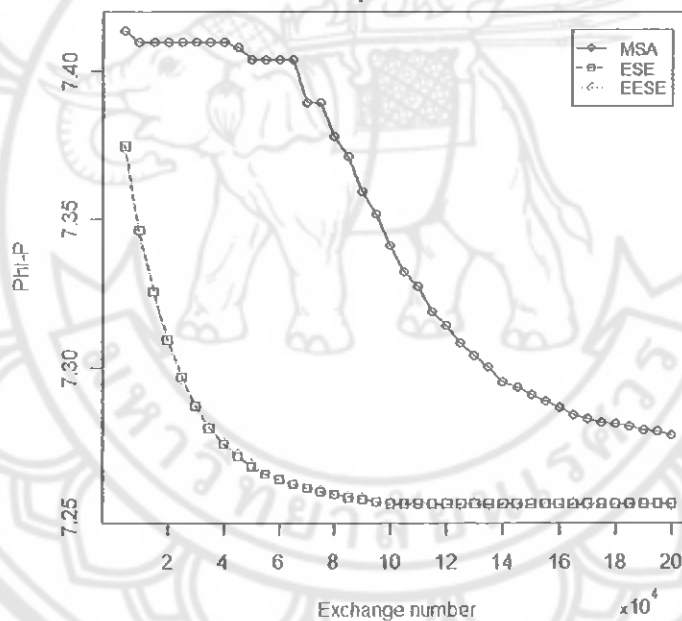
การเปรียบเทียบการลดลงของค่า  $\Phi_p$  ของแผนการทดลองขนาด  $201 \times 10$  โดยให้แต่ละอัลกอริทึมทำการแลกเปลี่ยนสมาชิกเพื่อสร้างแผนการทดลองใหม่จำนวน 200,000 รอบ จากแผนการทดลองเริ่มต้นที่แตกต่างกันโดยการสุ่ม กระทำทั้งหมด 10 ครั้งแล้วทำการหาค่าเฉลี่ยของจำนวนการครั้งแลกเปลี่ยนสมาชิก เพื่อพิจารณาการลดลงของค่า  $\Phi_p$



การลดของค่า  $\Phi_p$  ของแผนการทดลองขนาด  $201 \times 10$  ต่อจำนวนรอบการแลกเปลี่ยนสมาชิก

ภาพแสดงกราฟแสดงการลดลงของค่า  $\Phi_p$  โดยเริ่มต้นเมื่อแต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปจำนวน 5,000 รอบ ในแผนการทดลองขนาด  $201 \times 10$  แสดงให้เห็นว่าค่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้นจาก MSA ( $\Phi_p = 6.254$ ) ไม่สามารถเข้าใกล้ (Converge) แผนการทดลองที่เหมาะสมเมื่อผ่านการแลกเปลี่ยนสมาชิกไปประมาณ 100,000 รอบ ซึ่งตรงกันข้ามกับ ESE ( $\Phi_p = 6.172$ ) และ EESE ( $\Phi_p = 6.172$ ) และเมื่อปล่อยให้แต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปถึง 200,000 รอบ ค่า  $\Phi_p$  จากแผนการทดลองที่สร้างขึ้นโดย MSA ( $\Phi_p = 6.183$ ) ยังไม่สามารถลดต่ำลงได้เท่ากับ ESE ( $\Phi_p = 6.172$ ) และ EESE ( $\Phi_p = 6.172$ ) ซึ่งบ่งชี้ให้เห็นว่า กระบวนการค้นหาของ MSA ไม่สามารถสร้างแผนการทดลองที่เหมาะสมได้เมื่อแผนการทดลองมีขนาด  $201 \times 10$  และเมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้นโดย ESE ลดลงใกล้เคียงกับ EESE เมื่อจำนวนการแลกเปลี่ยนสมาชิกเพิ่มขึ้น

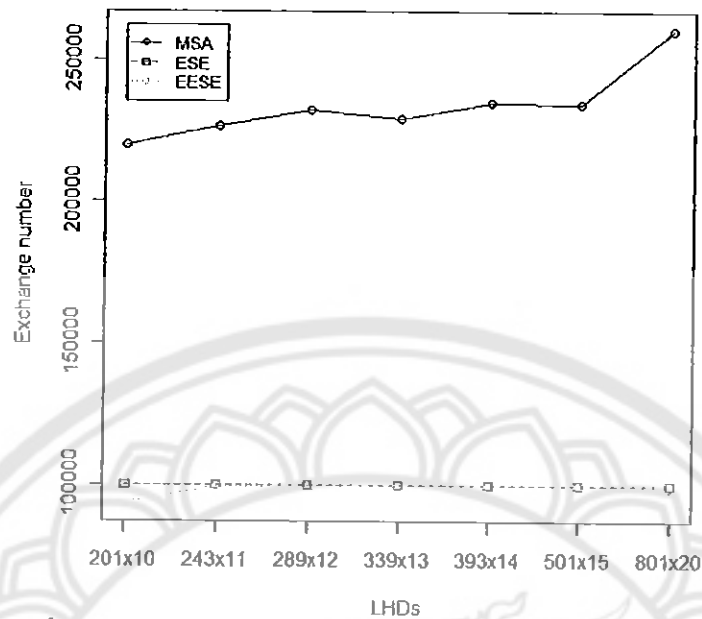
การเปรียบเทียบการลดลงของค่า  $\Phi_p$  ของแผนการทดลองขนาด  $801 \times 20$  โดยให้แต่ละอัลกอริทึมทำการแลกเปลี่ยนสมาชิกเพื่อสร้างแผนการทดลองใหม่จำนวน 200,000 รอบ จากแผนการทดลองเริ่มต้นที่แตกต่างกันโดยการสุ่ม กระทำทั้งหมด 10 ครั้งแล้วทำการหาค่าเฉลี่ยของจำนวนการครั้งแลกเปลี่ยนสมาชิก เพื่อพิจารณาการลดลงของค่า  $\Phi_p$



การลดของค่า  $\Phi_p$  ของแผนการทดลองขนาด  $801 \times 20$  ต่อจำนวนรอบการแลกเปลี่ยนสมาชิก

ภาพกราฟแสดงการลดลงของค่า  $\Phi_p$  ในแผนการทดลองขนาด  $801 \times 20$  โดยเริ่มต้นแสดงเมื่อแต่ละอัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปจำนวน 5,000 รอบ ผลที่ได้นั้นใกล้เคียงกับแผนการทดลองขนาด  $201 \times 10$  จากกราฟแสดงให้เห็นได้อย่างชัดเจนยิ่งขึ้นว่า MSA ( $\Phi_p = 7.277$ ) ไม่สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  ได้ต่ำเท่า ESE ( $\Phi_p = 7.254$ ) และ EESE ( $\Phi_p = 7.254$ ) เมื่ออัลกอริทึมผ่านการแลกเปลี่ยนสมาชิกไปถึง 200,000 รอบ ซึ่งตรงกันข้ามกับ ESE และ EESE ที่สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  ที่ต่ำเมื่อผ่านการแลกเปลี่ยนสมาชิกไปเพียง 100,000 รอบ

และ เมื่อพิจารณาเปรียบเทียบระหว่าง ESE และ EESE พบว่าค่า  $\Phi_p$  ของแผนการทดลองที่สร้างขึ้น จาก ESE และ EESE ลดลงเท่ากันเมื่อจำนวนการแลกเปลี่ยนสมาชิกเพิ่มขึ้น

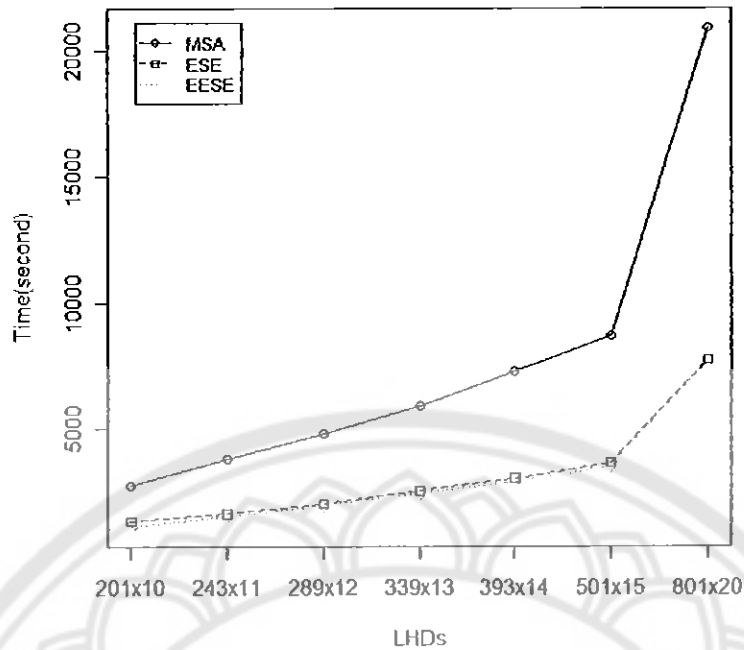


จำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมกับปัญหาขนาดใหญ่

การเปรียบเทียบจำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสม ปัญหาขนาดเล็ก โดยการพิจารณาที่ค่า  $\Phi_p$  ที่ต่ำที่สุดของแต่ละอัลกอริทึมหลังจบการทำงานจำนวน 10 ครั้ง แล้วจึงนำไปหาค่าเฉลี่ย

ภาพกราฟแสดงจำนวนรอบการแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมของ อัลกอริทึม MSA, ESE และ EESE ผลที่ได้คือในแผนการทดลองขนาด  $201 \times 10$  EESE ใช้จำนวนรอบ การแลกเปลี่ยนสมาชิกในการสร้างแผนการทดลองที่เหมาะสมเท่ากับ 904,070 ซึ่งน้อยกว่า ESE ที่ใช้ จำนวนรอบการแลกเปลี่ยนสมาชิกคือ 100,000 และ MSA ใช้จำนวนรอบการแลกเปลี่ยนสมาชิกมาก ที่สุดคือ 219,912 รอบ และจำนวนรอบการแลกเปลี่ยนสมาชิกของ ESE และ EESE จะเริ่มใกล้เคียงกัน เมื่อแผนการทดลองมีขนาดเพิ่มขึ้น โดยเท่ากันที่แผนการทดลองขนาด  $339 \times 13$  โดยใช้จำนวนรอบการ แลกเปลี่ยนสมาชิกคือ 100,000 ในขณะที่ MSA ใช้จำนวนรอบการแลกเปลี่ยนสมาชิกเท่ากับ 234,783 จากที่กล่าวมาสามารถสรุปได้ว่า MSA ใช้จำนวนรอบการแลกเปลี่ยนสมาชิกมากที่สุดในการสร้าง แผนการทดลองที่เหมาะสม และมากกว่า ESE และ EESE ถึง 2.6 เท่าในแผนการทดลองขนาด  $801 \times 20$  และเมื่อเปรียบเทียบระหว่าง ESE และ EESE พบว่าจำนวนรอบการแลกเปลี่ยนสมาชิกในการ สร้างแผนการทดลองที่เหมาะสมโดยเฉลี่ย EESE น้อยกว่า ESE

การพิจารณาเปรียบเทียบเวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมปัญหาขนาดใหญ่ โดยการพิจารณาที่ค่า  $\Phi_p$  ที่ต่ำที่สุดของแต่ละอัลกอริทึมหลังจบการทำงานจำนวน 10 ครั้ง แล้วจึงนำ เวลาที่ได้ไปหาค่าเฉลี่ย



### เวลาในการสร้างแผนการทดลองที่เหมาะสมกับปัญหาขนาดใหญ่

ภาพกราฟแสดงเวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมของ MSA, ESE และ EESE เมื่อพิจารณาพบว่า MSA ใช้เวลามากที่สุดในการสร้างแผนการทดลองที่เหมาะสมสำหรับแผนการทดลองขนาด  $801 \times 20$  โดยใช้เวลาเท่ากับ 20854.01 วินาที รองลงมาคือ ESE ที่ใช้เวลาเท่ากับ 7744.424 วินาที และ EESE ที่ใช้เวลาเท่ากับ 7672.487 วินาทีตามลำดับ จะเห็นว่า ESE และ EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมสำหรับปัญหา  $801 \times 20$  ใกล้เคียงกัน สรุปได้ว่า เมื่อแผนการทดลองมีจำนวนตัวแปรเข้ามากกว่า 10 ตัวแปร MSA ไม่สามารถสร้างแผนการทดลองที่เหมาะสมได้ในเวลาที่สามารถยอมรับได้ โดยใช้เวลามากกว่า ESE และ EESE เท่ากับ 2.5 เท่าในแผนการทดลองขนาด  $801 \times 20$  และ ESE และ EESE เวลาที่ใช้ในการสร้างแผนการทดลองที่เหมาะสมเริ่มใกล้เคียงกันเมื่อขนาดของแผนการทดลองใหญ่ขึ้น

จากการทดสอบอัลกอริทึมการสืบค้นกับปัญหาขนาดใหญ่ สามารถสรุปได้ว่า MSA เป็นอัลกอริทึมที่ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมนานกว่าอัลกอริทึม ESE และ EESE อีกทั้งกระบวนการค้นหาของ MSA ยังไม่สามารถสร้างแผนการทดลองที่มีคุณสมบัติที่ใกล้เคียงกับ ESE และ EESE ( $\Phi_p$  ใกล้เคียงกัน) ในขณะที่ ESE และ EESE สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  เท่ากัน แต่ EESE ใช้เวลาในการสร้างแผนการทดลองที่เหมาะสมน้อยกว่า ESE ทำให้สามารถสรุปได้ว่า เมื่อแผนการทดลองมีจำนวนตัวแปรเข้าตั้งแต่ 10 ตัวแปร อัลกอริทึมที่เหมาะสมในการสร้างแผนการทดลองที่ให้คุณสมบัติที่ดีคือ EESE

## บทที่ 5

### สรุปผลการวิจัย

#### สรุปผลการวิจัย

จากการวิจัยในครั้งนี้ อัลกอริธึมการสืบค้นที่แบบสโทแคสติกเชิงวิวัฒนาการที่พัฒนาขึ้นสำหรับการสร้างแผนการทดลองที่เหมาะสม โดยการนำข้อดีจากอัลกอริธึม MSA และ ESE มารวมเข้าด้วยกัน จากผลการวิจัย สามารถสรุปได้ดังนี้

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

2. EESE มีประสิทธิภาพสูงกว่า MSA เมื่อพิจารณาที่จำนวนการแลกเปลี่ยนสมาชิก และเวลาที่ใช้ในการทำงานในหน่วยวินาที และ EESE สามารถสร้างแผนการทดลองที่ให้ค่า  $\Phi_p$  น้อยกว่า MSA กับปัญหาขนาดใหญ่

3. EESE มีประสิทธิภาพสูงกว่า ESE เมื่อพิจารณาที่จำนวนการแลกเปลี่ยนสมาชิกในปัญหขนาดเล็ก แต่ในปัญหาขนาดใหญ่ EESE ไม่สามารถสร้างแผนการทดลองที่เหมาะสมได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยกว่า ESE มากนัก สืบเนื่องมาจากกฎการหยุดสืบค้นไม่สามารถทำให้ EESE หยุดทำงานได้ในจำนวนรอบการแลกเปลี่ยนสมาชิกที่น้อยลง ซึ่งจากการศึกษาพบว่า เมื่อปัญหามีขนาดใหญ่แผนการทดลองสุดท้ายที่ได้ในแต่ละรอบการทำงานจะให้ค่า  $\Phi_p$  ที่น้อยลงเสมอในทุกๆ รอบสรุปคือ EESE มีประสิทธิภาพสูงกว่า ESE เล็กน้อยเมื่อพิจารณาเปรียบเทียบเวลาในการสร้างแผนการทดลองทั้งกับปัญหขนาดเล็กและปัญหาขนาดใหญ่

4. คุณสมบัติของแผนการทดลองที่วัดโดยใช้เกณฑ์การเลือกค่าเหมาะสม  $\Phi_p$  โดยเฉลี่ยแล้ว MSA, ESE และ EESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ใกล้เคียงกันในปัญหขนาดเล็ก แต่กับปัญหาขนาดใหญ่ ESE และ EESE สามารถสร้างแผนการทดลองที่เหมาะสมได้ดีกว่า MSA

#### ข้อเสนอแนะ

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

2. ควรพัฒนากฎการหยุดสืบค้นให้เหมาะสมมากขึ้น โดยการประยุกต์ใช้วิธีการทางสถิติ เพื่อให้ให้อัลกอริทึมเสียจำนวนรอบการทำงานโดยไม่มีประโยชน์

3. ควรนำแผนการทดลองที่เหมาะสมที่ได้จากการสร้างของ EESE และ ESE ไปพัฒนาโมเดลเพื่อการพยากรณ์ เพื่อวัดประสิทธิภาพของโมเดลที่ได้จากแผนการทดลองว่ามีความแม่นยำเพียงใด



## บรรณานุกรม

- Boonjarenpol, B. (2005). **Simulation**. Retrieved September 13, 2008, from [http://www.krirk.ac.th/education/dr\\_boonserm/simulation.pdf](http://www.krirk.ac.th/education/dr_boonserm/simulation.pdf)
- Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimal conditions. **Journal of the Royal Statistical Society**, 13, 1-38.
- Chambers, J. and Colleagues. (2009). **R Foundation for statistical computing**. Retrieved September 13, 2008, from <http://www.r-project.org>
- Cressie, N. (1989). Geostatistics. **The American Statistician**, 43(4), 197-202.
- Dorigo , M. and Gambardella, L.M. (1997). Ant colonies for the traveling salesman Problem. **Biosystems. Elsevier Science**, 43(2), 73-81.
- Fang, K.T. Lin, D. K. J. Winker, P. and Zhang, Y. (2000). Uniform design: theory and application. **Technometrics**, 42, 237-248.
- Fang, K. T. (1980). Experimental design by uniform distribution. **Acta Mathematicae Applicatae Sinica**, 3, 363-372.
- Glover, F. (1986). Future path for integer programming and links to artificial intelligence. **Computer and operation research**, 13, 533-549.
- Goldberg, DE. (1989). **Genetic algorithms in search, optimization and machine learning**. MA: Addison-Wesley.
- Holland, J. (1975). **Adaptation in natural and artificial systems**. MI: University of Michigan Press.
- Jin, R. Chen, W. and Sudjianto A. (2005). An efficient algorithm for construct optimal design of computer experiment. **Journal of statistical Planning and Inference** 134, 268-287.
- Kirkpatrick, S. Gelatt, C.D. and Vecchi M.P.(1983). Optimization by simulated Annealing. **Science**, 220, 621-630.
- Koehler, J. R. and Owen, A. B. (1996). Computer experiments. In Ghosh, S. and Rao, C.R. (Eds.), **Handbook of Statistics**. (pp. 261-308), New York: Elsevier Science.
- Li, W. and Wu, C. F. J. (1997). Columnwise - pairwise algorithms with applications to the construction of supersaturated designs. **Technometrics**, 39, 171-179.
- McKay, M. D. Beckman, R. J. and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. **Technometrics**, 21(2), 239-245.





- Montgomery, D.C. (2005). **Design and analysis of experiments** (6<sup>th</sup> ed.). New York: John Wiley & Sons.
- Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational Experiments. **Journal of Statistical Planning and Inference**, 43, 381-402.
- Na-udom, A. (2008). Exploratory search algorithms for computer simulated experiments. **Burapha Science Journal**, 13(2), 106-120.
- Na-udom, A. (2008). Review of design and analysis of computer simulated experiments. **KKU Science Journal**, 16(2), 106-120.
- Na-udom, A. and Rungrattanaubol, J. (21-23 May 2008). **Surrogate models for response from computer simulated experiments. In Conference on Statistics and Its Applications**. Chonburi: A One The Royal Cruise Pattaya Hotel.
- Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and Visualization. **Statistica Sinica**, 2, 439-452.
- Pongcharoen, P. Stewardson, D.J. Hicks, C. and Braiden, P.M. (2001). Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry. **Journal of Applied Statistics**, 28(3), 441-455.
- Rungrattanaubol, J. and Na-udom, A. (2007). Comparison of evolutionary search algorithms in computer simulated experiments. **National Computer Science and Engineer Conference (NCSEC2007)**, 11, 102-106.
- Rungrattanaubol, and J. Na-udom, A. (21-23 May 2008). **Efficient methodology to construct optimal designs for computer simulated experiments. In Conference on Statistics and Its Applications**. Chonburi: A One The Royal Cruise Pattaya Hotel.
- Rungrattanaubol, J. and Na-udom, A. (September 3 – 4 2009). Stopping rules for search algorithm in the construction of the optimal designs. **National Operations Research Conference**. Bangkok: The Emerald Hotel.
- Saab, Y. G. and Rao, Y. B. (1991). Combinatorial optimization by stochastic evolution. **IEEE Transaction on Computer-aided Design**, 10, 525–535.
- 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.
- Simpson, T. W. Lin, D. K. J. and Chen, W. (2000). Sampling strategies for computer experiment: design and analysis. **International Journal of Reliability and Applications**, 2(3), 209-240.



- 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, 129-150.
- Tang, B. (1993). Orthogonal array-Based latin hypercubes. **Journal of the American Statistical Association**, 88(424), 1392-1397.
- Thamma, T. Rungtattanaubol, J. and Na-udom, A. (May 15-24 2008). Modification on search algorithm for computer simulated experiment. In **Proceedings of the National Operations Research Network Conference** (pp. 117 – 123), Bangkok: Chaopraya Park Hotel.





## ภาคผนวก

The 6th National Conference on Computing and Information Technology ที่จัดขึ้น ณ โรงแรมอีส  
เทิร์น กรุงเทพมหานคร ในวันที่ 3-5 มิถุนายน 2553







THE 6<sup>TH</sup> NATIONAL CONFERENCE ON  
COMPUTING AND INFORMATION  
TECHNOLOGY

PROCEEDINGS OF NCCIT 2010

THE 6<sup>TH</sup> NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

3 - 5 JUNE 2010, EASTIN HOTEL, BANGKOK, THAILAND  
WWW.NCCIT.NET

FACULTY OF INFORMATION TECHNOLOGY  
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

VOLUME 1

บทความวิจัย

การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
ครั้งที่ 6

3 - 5 มิถุนายน 2553 ณ โรงแรมอีสติน กรุงเทพฯ



คณะเทคโนโลยีสารสนเทศ  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ







THE 6<sup>TH</sup> NATIONAL CONFERENCE ON  
COMPUTING AND INFORMATION  
TECHNOLOGY

PROCEEDINGS OF NCCIT 2010

THE 6<sup>TH</sup> NATIONAL CONFERENCE ON COMPUTING AND INFORMATION  
TECHNOLOGY

3 - 5 JUNE 2010 : EASTIN HOTEL, BANGKOK, THAILAND  
WWW.NCCIT.NET

FACULTY OF INFORMATION TECHNOLOGY  
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

บทความวิจัย

การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
ครั้งที่ 6

3 - 5 มิถุนายน 2553 ณ โรงแรมอีสติน มหิทธิกัน กรุงเทพฯ ฯ



คณะเทคโนโลยีสารสนเทศ  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ



<b>Thursday June 3, 2010</b>		
8:00-9:00	Registration (Eastin Hotel Makkasan, Bangkok, Thailand)	
9:00-9.30	Opening Ceremony by Prof. Dr. Teravuti Boonyasopon, President of King Mongkut's University of Technology North Bangkok; Chairman, TOT Board of Directors	
9:30-10:30	Invited Keynote Speech by Prof. Dr. Dr. Wolfgang A. Halang	
	<i>Topic: Real-time Translation of Spoken Language</i>	
10:30-11:00	<i>Coffee Break</i>	
11:00-12:00	Invited Keynote Speech by Prof. Dr. Jesse Jin	
	<i>Topic: Web 2.0 Versus Enterprise 2.0 – How Communities Influence Today's Company Culture</i>	
12:00-13:00	<i>Lunch</i>	
<b>Room I</b>		
<b>Session 1: English Papers</b>		
<i>Chair: Dr. Gareth Clayton</i>		
<b>Time/Paper-ID</b>	<b>Title/Author</b>	<b>Page</b>
13:00-13:20 NCCIT2010-57	<b>Enhancement of Enhanced Stochastic Evolutionary Algorithm for Computer Simulated Experiment</b> <i>Thanyadol Chantarawong, Jaratsri Rungrattanaubol, and Anamai Na-udom</i>	3
13:20-13:40 NCCIT2010-104	<b>Applying Web 2.0 Technologies to eLearning: Opportunities and Challenges</b> <i>Brendan McCarthy</i>	9
13:40-14:00 NCCIT2010-196	<b>Fuzzy Rule Based System Generated by a Hybrid Heuristic Algorithm</b> <i>Adisak Sangsongfa and Phayung Meesad</i>	14
14:00-14:20 NCCIT2010-191	<b>A New Feature Selection Method for Gene Expression Data</b> <i>Patharawut Saengsiri, Sageemas Na Wichian, and Phayung Meesad</i>	20
14:00-14:40 NCCIT2010-222	<b>Q-Map Application for Enrichment of a Mobile Directory Assistance Service</b> <i>Mohammed Samaka, Adnan Abu-Dayya, B. Owaisho, N. Al Sada, N. Al Khuzaai, and R. Al Ashqar</i>	26
14:20-14:40 NCCIT2010-18	<b>Using Remote Sensing To Detect Change land surface features</b> <i>Amro F. Alasta</i>	31
14:40-15:00	<i>Coffee Break</i>	



NCCIT2010-57

## Enhancement of enhanced stochastic evolutionary algorithm for computer simulated experiment

Thanyadol Chantarawong<sup>1</sup>, Jaratsri Rungrattanaubol<sup>2,c</sup> and Anamai Na-udom<sup>3</sup>

<sup>1</sup>Master student, Department of Computer Science and Information Technology,

<sup>2,c</sup>Lecturer, Department of Computer Science and Information Technology,

<sup>3</sup>Lecturer, Department of Mathematics

Naresuan University

rockwalnut@gmail.com, jaratsrir@nu.ac.th, anamain@nu.ac.th

### Abstract

Computer simulated experiments (CSE) are currently used to replace complex engineering systems and applications. The design of experiment plays the major roles in CSE, since the accurate surrogate model usually obtains from the best experimental design. The process of constructing the design in this context is a space filling process, so search techniques along with different optimal criterion are applied [1],[2],[3]. This paper purposes the enhancement of enhanced stochastic evolutionary (ESEE), adapted from the enhanced stochastic evolutionary (ESE), to search or construct the optimal latin hypercube design (LHD) based on  $\Phi_p$  criteria. The results show ESEE performs better than ESE in any dimensional designs, especially in a small dimension in terms of time and number of element-exchange.

**Keyword:** Computer simulated experiment, Stochastic evolutionary, Optimal design, Latin hypercube design.

### 1. Introduction

Recently computer simulated experiment (CSE) has been widely used in a complex engineering system to analyse the performance of a system. This is probably because some physical experiments have many limits such as very high cost of instruments or too advanced and high technology to run the experiments. CSE consists of 3 major parts; experimental design,

simulation routine and approximation model as presented in Figure 1.

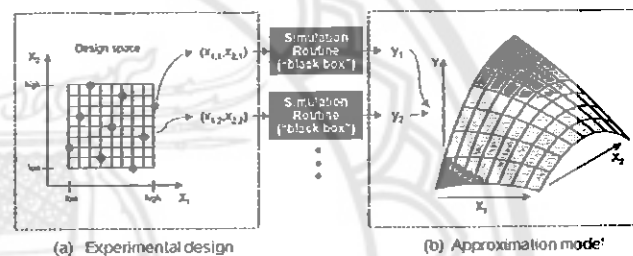


Figure 1: Three parts of CSE [1]

The inspiration of CSE is to obtain an approximation model with high prediction accuracy. In order to achieve this, we require an optimal design, which is efficient enough to represent a design sample from a minimal sampling point (run), or the design with high space filling properties (a coverage of design space). The model built from the optimal design is guaranteed to be highly accurate.

CSE have widely attracted many researchers. Sack et al. [2] introduced and demonstrated that CSE is correlated with the computational searches for an optimal design and the statistical model building. McKay et al. [3] proposed Latin hypercube design (LHD) for use in CSE. Morris and Mitchell [4] adopted a version of a simulated annealing (SA) along with  $\Phi_p$  criterion for constructing an optimal design. Thamma et al. [5] presented a modified version of SA to improve the performance of an original SA by adding a tolerance level in a replacement design process. Jin et al. [6]



developed Enhanced stochastic evolutionary (ESE) algorithm for searching an optimal design.

An experimental design or so called sampling is a set of sampled value of input variable from a design space. An objective of searching for an optimal design is to obtain a design that meets a space filling property so that the model built from this design will be very accurate. In Figure 1, the input variable  $X_1$  and  $X_2$  were sampled with 9 points of run. The 9 points of run were given as an input to a simulation routine (or “black box”). The output ( $Y$ ) together with the 9 points of run are formed an approximation model (or a surrogate model).

Simulation routine is typically high fidelity, high cost and takes a long time, since the code is very complex and implemented by an expert. Thus, the design of experiment plays a crucial role here. The question is how we can search for optimal LHDs from the space with size  $(n!)^d$ , where  $n$  is the number of run and  $d$  is the number of input variables.

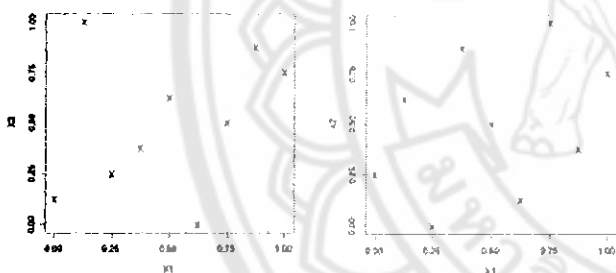


Figure 2: 9x2 LHD before and after applying search algorithm

CSE is very complex and contains a lot of input variables. In nature, a design space is very large. To search for an optimal design from a large space, many search algorithms are applied along with various optimality criteria used as objective or fitness functions. Examples of LHD for CSE are shown in Figure 2, the LHD on the right hand side is an example of an optimal LHD constructed from a search algorithm based on  $\Phi_p$  criterion, more details in section 2.5.

The aim of this paper is to develop a search technique named Enhanced stochastic evolutionary (ESE) introduced by Jin et al. [6] to construct an optimal LHD

based on  $\Phi_p$  criterion and to improve its performance. The comparative study will be conducted to present how enhancement of ESE (EESE) performs better, especially with a large dimension of design, e.g. 801x20, in terms of computational time and number of exchange elements.

## 2. Research methodology

We first develop the original ESE and later adapt the EESE using R language. The class of design used here is Latin hypercube design (LHD) along with  $\Phi_p$  criterion.

### 2.1 Latin hypercube design (LHD)

LHD is widely used in CSE in a part of experimental design. LHD is  $n \times d$  matrix ( $n$  runs and  $d$  input variables), each column contains 1, 2, 3, ...,  $n$  run, randomly permutation from possible values of input variable, which is divided into an equal interval and takes a form of a unit interval [0,1] as shown in Figure 3.

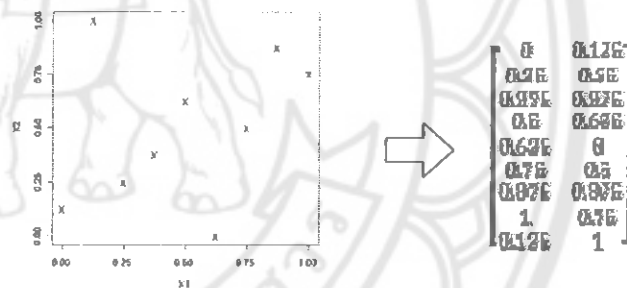


Figure 3: 9x2 Random LHD

In Figure 3, it is 9x2 LHD, which the 1<sup>st</sup> column represents the input value for 9 runs of input variable  $X_1$  and the 2<sup>nd</sup> column represents  $X_2$ . The value of each input variable is in the range of [0, 1] with 9 equal interval values. Each row represents a point of run. LHD is flexible since it can apply for a design with any number of input variables and any number of run. This characteristic can ensure the stratified sampling technique [2].

### 2.2 Element-exchange operation

The element exchange operation to construct a new LHD design is developed from a concept of column-wise operation purposed by Li and Wu [7]. The process





is randomly interchange two distinct elements in a column as shown in Figure 4. After element-exchange, the design maintains LHD properties.

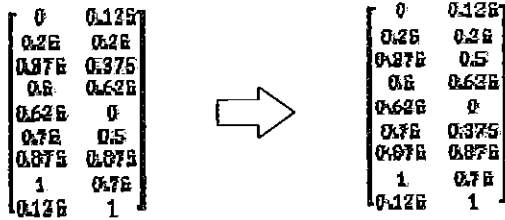


Figure 4: Element exchange in the 2<sup>nd</sup> column of a 9x2 LHD

### 2.3 Enhanced stochastic evolutionary (ESE)

ESE is developed from Stochastic evolutionary (SE) algorithm proposed by Sabb and Rao [8], contains 2 nested loop called *inner* and *outer* loop. The inner loop performs a local search process by constructing a new design and decides for an acceptance of this new design. The outer loop over the inner loop performs a global search controlled by adjusting the threshold ( $T_h$ ) base on *acceptance ratio* and *improved ratio* from the inner loop.

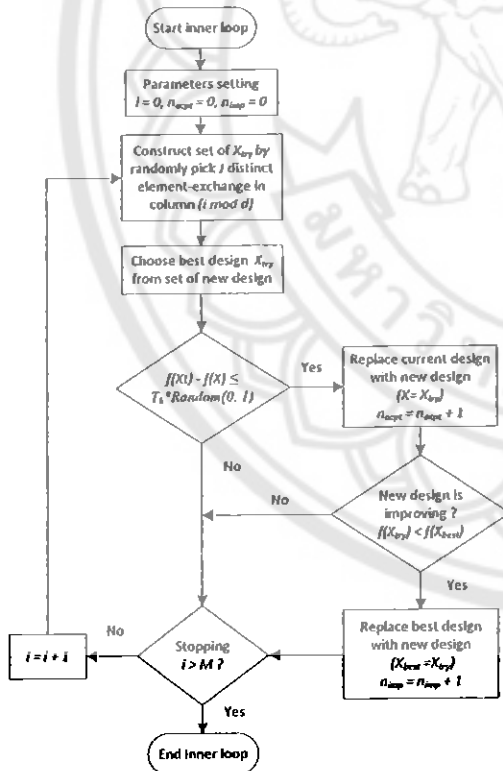


Figure 5: Flowchart of ESE inner loop [6]

#### Inner loop process (Figure 5)

Step 1: after initializing parameters and design  $X_0$  in the outer loop,  $X = X_0$ ,  $i = 0$ ,  $n_{accept} = 0$  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 5.

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: end the inner loop if  $i > M$  else go to step 2.

The parameters  $J$  and  $M$  in this study is  $j = \binom{n}{2}/5$  but not larger than 50 and  $M = 2 \binom{n}{2} d / J$  but not larger than 100.

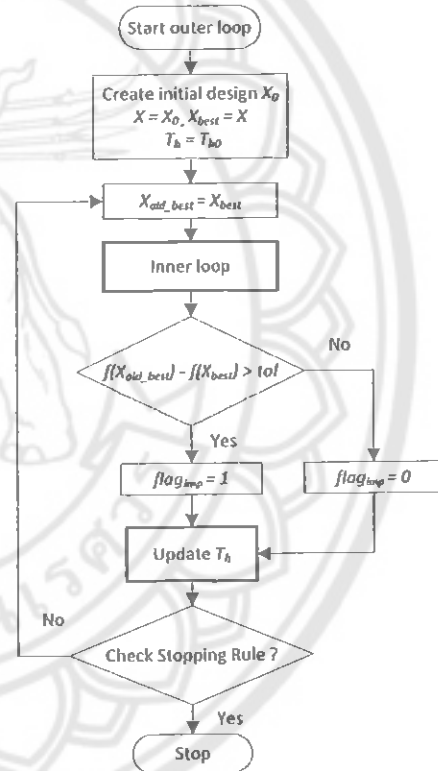


Figure 6: Flowchart of ESE outer loop [6]

#### Outer loop process (Figure 6)

Step 1: randomly create an initial design  $X_0$  and set  $X = X_0$ ,  $X_{best} = X$ , initialize  $T_{h0} = 0.005 * \Phi_p(X_0)$  and  $T_h = T_{h0}$ .

Step 2: let  $X_{old\_best} = X_{best}$ .

Step 3: go to the inner loop process.

Step 4: select a method to update  $T_h$ , by setting  $flag_{imp}$ .



Step 5: update  $T_h$ , more details later.

Step 6: terminated by a stopping rule else goto step 2.

The tolerance  $tol$  in this paper is set to 0.0001; from the empirical study 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 named *improving process* and *exploration process*. The search process is in 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 to help in searching for the best in a local optimal design based on an accept ratio ( $n_{acpt}/M$ ) and improve ratio ( $n_{imp}/M$ ).

- If  $n_{acpt}/M > \beta_1$  and  $n_{imp}/M < n_{acpt}/M$ , then  $T_h$  is decreased by equation  $T_h = \alpha_1 * T_h$ .
- If  $n_{acpt}/M > \beta_1$  and  $n_{imp}/M = n_{acpt}/M$ , then  $T_h$  is unchanged.
- Otherwise,  $T_h$  is increased by equation  $T_h = T_h/\alpha_1$ ,

where  $0 < \alpha_1 < 1$  and  $0 < \beta_1 < 1$ , we suggest  $\alpha_1 = 0.8$  and  $\beta_1$  should be small we suggest  $\beta_1 = 0.1$ [6].

In *exploration process* ( $flag_{imp} = 0$ ).  $T_h$  will be adjusted to help algorithm move far away from a local optimal design based on the range of accept ratio.

- If  $n_{acpt}/M < \beta_2$ , then  $T_h$  is increased till  $n_{acpt}/M > \beta_3$  by equation  $T_h = T_h/\alpha_3$ .
- If  $n_{acpt}/M > \beta_3$ , then  $T_h$  is decreased till  $n_{acpt}/M < \beta_2$  by equation  $T_h = \alpha_2 * T_h$ ,

where  $0 < \alpha_2 < \alpha_3 < 1$  and  $0 < \beta_2 < \beta_3 < 1$ , we suggest  $\alpha_2 = 0.9$ ,  $\alpha_3 = 0.7$ . While  $\beta_2$  should be small, we set  $\beta_2 = 0.1$  and  $\beta_3$  should be large enough,  $\beta_3 = 0.8$  [6].

### 2.4 Enhancement of ESE (EESE)

EESE is modified from ESE by combining the advantage of SA (i.e. local search process) and the advantage of ESE (i.e. global search process). EESE contains 2 nested loop similarly to ESE as displayed in Figure 7. The outer loop is almost the same as in ESE, there is only a change in a stopping rule 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  $X_{globalbest}$   $\delta$  consecutive times, then the search process will terminate. In this study we set  $\delta$  to 10.

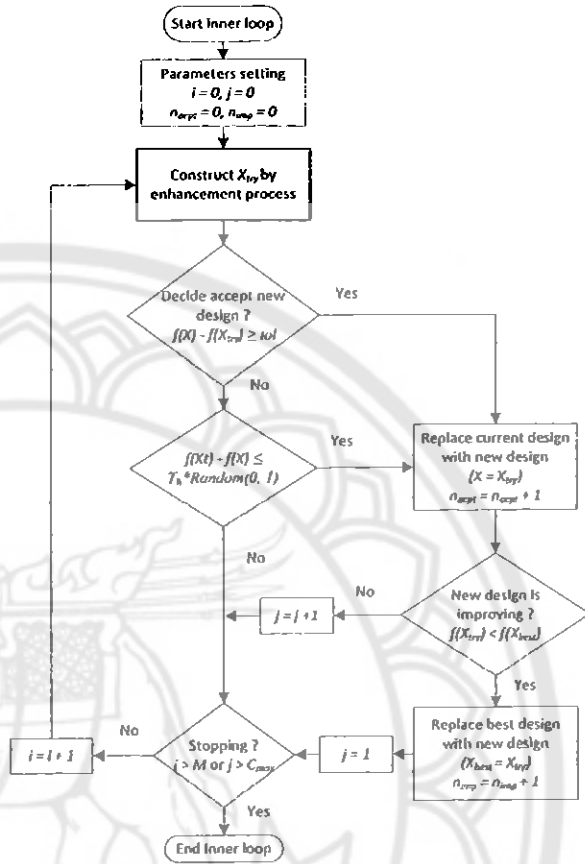


Figure 7: Flowchart of EESE inner loop

In the inner loop, there are many changes. Step 2, step 5 and step 6 are changed. In step 2, the process of constructing a new design  $X_{ij}$ , is changed to element-exchange in column  $(i \text{ mod } d)$  for all  $J$  iterations while the original ESE randomly picks  $J$  distinct element-exchange in column  $(i \text{ mod } d)$ . Then the computational complexity decreases from  $O(n^2)$  to  $O(n)$ .

It is proven by; in ESE a random element-exchange process for all  $J$  iteration, then in every iteration  $i$  musts check for a distinct  $i-1$  loop, hence the complexity is  $O(n*(n-1)) = O(n^2)$ . In EESE, we adapted the process of element-exchange from SA shown in Figure 8.  $J$  iteration element-exchange of a current design  $X$  in column  $i \text{ mod } d$  is independent. It is no need to perform all  $J$  iterations. The computation complexity becomes  $O(n)$ .



In step 5, if a new design  $X_{ny}$  is improved to be better than the best 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 use  $C_{max} = 10$ .

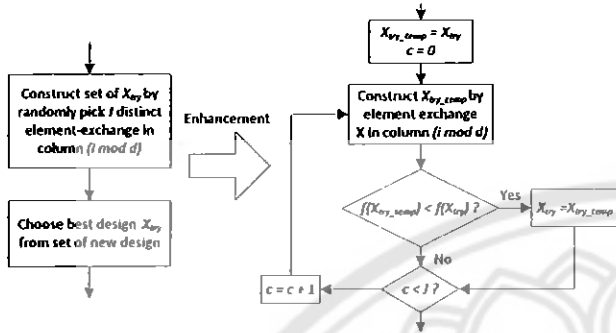


Figure 8: A new design construction using SA (Step 2)

## 2.5 $\Phi_p$ Criterion

In this study, the  $\Phi_p$  criterion, proposed by Morris and Mitchell [4], is used to evaluate the design. It is developed from a maximin distance criterion. Given design  $X$ , the Euclidean inter-site distance between the sample points is calculated from equation (1).

$$d(x_j, x_k) = \left[ \sum_{i=1}^k (x_{ji} - x_{ki})^2 \right]^{1/2} \quad (1)$$

From Euclidean distance for  $X$ , we get a symmetric matrix  $D = [d_{j,k}]_{n \times n}$ . Then  $\Phi_p$  is calculated by equation (2).

$$\Phi_p = \left[ \sum_{i=1}^n \sum_{j=i+1}^n (1/d_{ij})^p \right]^{1/p}; p \in I^+ \quad (2)$$

Equation (2) is a direct method to compute  $\Phi_p$ , which takes a long time. Jin et al [6] noticed that in the process of element-exchange in ESE, only some values in matrix  $D$  are changed, so it is no need to recalculate the complete matrix  $D$ . In element-exchange operation, after exchanging elements of a design  $X$  between rows  $i_1$  and  $i_2$  within column  $k$  ( $x_{i_1k} \leftrightarrow x_{i_2k}$ ), there are changes only elements in row  $i_1$  and  $i_2$  and column  $i_1$  and  $i_2$  in matrix  $D$  [6]. For any  $1 \leq j \leq n$  and  $j \neq i_1, i_2$  let

$$s(i_1, i_2, k, j) = |x_{i_1k} - x_{jk}| - |x_{i_2k} - x_{jk}|$$

$$\text{then } d'_{i_1j} = d'_{j i_1} = [d'_{i_1j} + s(i_1, i_2, k, j)]^{1/p}$$

$$\text{and } d'_{i_2j} = d'_{j i_2} = [d'_{i_2j} - s(i_1, i_2, k, j)]^{1/p}$$

The simplified version of  $\Phi_p$  calculation is:

$$\Phi_p' = \left[ \Phi_p^p + \sum_{j=1}^n [(d'_{i_1j})^{-p} - (d_{i_1j})^{-p}] + \sum_{j=1}^n [(d'_{i_2j})^{-p} - (d_{i_2j})^{-p}] \right]^{1/p} \quad (3)$$

where  $j \neq i_1, i_2$ , and  $p = 5$  and  $l = 2$ . This version of  $\Phi_p$  has a significant reduction of computational time comparing to  $\Phi_p$  calculation in equation (2) [6].

## 2.6 Data quality

We compared ESE with ESE to construct optimal LHDs for a given dimension of design shown in Table 1.

Table 1: Dimension of LHDs

$d$	2	5	10	15	20
$n$	9	51	201	451	801

For each dimension of LHDs, we run ESE and ESE for 10 times on a PC with AMD Athlon 64 X2 3.0 GHz CPU. The results in terms of *efficiency* are timed in second and number of element-exchange to construct optimal LHDs with the minimization of  $\Phi_p$  criterion.

## 3. Results and comparative studies

The results in terms of  $\Phi_p$  criterion values for each dimension of LHDs obtained from ESE and ESE are presented in Table 2. In Table 2,  $\Phi_p$  values from ESE are slightly lower than ESE when the dimension is small whereas ESE is superior to ESE when the dimension becomes larger. For large dimension of designs, these two algorithms perform similar results of  $\Phi_p$  values, indicates that either ESE or ESE can be applied.

The results based on the performance (*efficiency*) for ESE and ESE are presented in Table 3. When the dimension of design is small, ESE converges much faster than ESE and the number of exchanges required in the search process is less than the ESE. For a large dimension of design, the number of exchange is fixed at 100000 ESE converges slightly faster than ESE as



shown in Figure 9. This indicates if time constraint is taken into account, EESE seems to be the better choice to use in the construction of the optimal LHD designs.

**Table 2:** The values of  $\Phi_p$  criterion from ESE and EESE

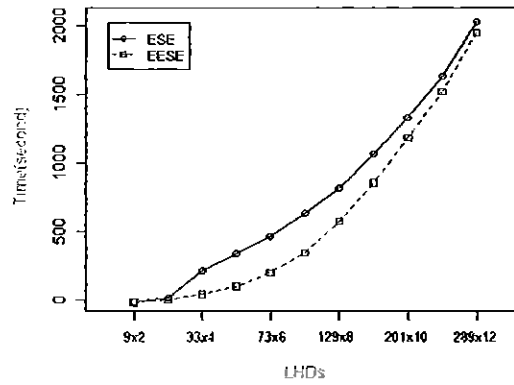
LHDs	Algorithm	$\Phi_p (p = 5, l = 2)$			
		Min	Max	Mean	SD
9 × 2	ESE	4.273	4.344	4.300	0.035
	EESE	4.273	4.364	4.317	0.038
51 × 5	ESE	5.424	5.439	5.432	0.005
	EESE	5.427	5.439	5.433	0.004
201 × 10	ESE	6.170	6.174	6.172	0.001
	EESE	6.171	6.173	6.172	0.000
451 × 15	ESE	6.760	6.762	6.761	0.000
	EESE	6.760	6.761	6.761	0.000
801 × 20	ESE	7.253	7.254	7.254	0.000
	EESE	7.253	7.254	7.254	0.000

**Table 3:** Performance of ESE and EESE

LHDs	Algorithm	Performance (Average)		Time ESE/EESE
		Time (sec.)	#Exchange	
9 × 2	ESE	1.829	2880	1.866
	EESE	0.98	2281	
51 × 5	ESE	358.976	100000	3.056
	EESE	117.428	44080	
201 × 10	ESE	1349.488	100000	1.119
	EESE	1205.082	96915	
451 × 15	ESE	3648.646	100000	1.044
	EESE	3492.132	100000	
801 × 20	ESE	7744.424	100000	1.009
	EESE	7672.487	100000	

#### 4. Conclusions

This paper presents the method to enhance the ESE to construct the optimal LHD for use in CSE. The major enhancement appears in the inner loop as shown in Figure 7 and 8. As presented in the previous section, EESE perform better than ESE in terms of efficiency. Therefore EESE would be recommended in the construction of LHD for use CSE. In order to extend and complete the conclusion, the validation of the surrogate model accuracy could be further investigated.



**Figure 9:** Time used in ESE and EESE for various dimensions

#### References

- [1] T.W. Simpson, D. K. J. Lin, and W. Chen, "Sampling Strategies for Computer Experiment: Design and Analysis," *International Journal of Reliability and Applications*, Vol. 2, No. 3, pp. 209-240, 2000.
- [2] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp. 409-435, 1989.
- [3] M. D. McKay, 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, No. 2, pp. 239-245, 1979.
- [4] M. D. Morris, and T. J. Mitchell, "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, Vol. 43, pp. 381-402, 1995.
- [5] T. Thamma, J. Rungtattanabool, and A. Na-udom, "Modification on Search Algorithm for Computer Simulated Experiment," *Proceedings of national operations research conference*, pp. 117 – 123, May 15th-24th, Thailand, 2008.
- [6] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for construct optimal design of computer experiment," *Journal of statistical Planning and Inference*, Vol. 134, pp. 268-287, 2005.
- [7] W. Li, and C. F. J. Wu, "Columnwise-Pairwise Algorithms with Applications to the Construction of Supersaturated Designs," *Technometrics*, Vol. 39, pp. 171-179, 1997.
- [8] Y. G. Saab, and Y. B. Rao, "Combinatorial optimization by stochastic evolution," *IEEE Transaction on Computer-aided Design*, Vol.10, pp. 525-535, 1991.









๑๖๖๖๖๖๖๖

๑๑๑๑

