



สำนักหอสมุด

## อภิธาน์ทนาการ

การออกแบบการทดลองคอมพิวเตอร์โดยใช้อัลกอริทึมการสืบค้น

Design for computer experiment using search algorithms

จรัสศรี รุ่งรัตนอุบล  
เกตุจันทร์ จำปาไชยศรี

สำนักหอสมุด มหาวิทยาลัยนครสวรรค์  
วันลงทะเบียน.....5...VII...2011...  
เลขทะเบียน...15648099...  
เลขเรียกหนังสือ...จ 0ค

๑.ค.๔3  
๖1๖๓  
2550

รายงานงานวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยนครสวรรค์  
งบประมาณวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยนครสวรรค์ประจำปี 2550

## ประกาศคุณูปการ

งานวิจัยนี้สำเร็จลงได้ด้วยดีด้วยความกรุณาอย่างยิ่งจาก คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร  
ที่ได้ให้การสนับสนุนเงินวิจัย

ขอขอบคุณ ดร. เกตุจันทร์ จำปาไชยศรี ผู้ร่วมวิจัยที่ให้คำแนะนำเกี่ยวกับตัวชี้วัดทาง  
สถิติ และขอขอบคุณ ดร. อนามัย นาอุดม ที่ให้คำแนะนำในเรื่องการออกแบบการทดลองด้วย  
คอมพิวเตอร์ ตลอดจนให้ความร่วมมือในการเขียนบทความเพื่อนำเสนอในงานประชุมวิชาการ  
NCSEC2007 ที่จัดขึ้นที่ โรงแรมมิราเคิล แกรนด์ จ. กรุงเทพฯ ในวันที่ 19-21 พฤศจิกายน 2550

นอกจากนี้ ข้าพเจ้าต้องขอขอบใจ นิสิตช่วยวิจัยทุกคนที่ร่วมแรงร่วมใจพัฒนาโปรแกรมจน  
สำเร็จเป็นอย่างดี

คุณค่าและประโยชน์อันพึงมีจากงานวิจัยนี้ ผู้จัดทำขอขอบอุทิศแด่ผู้มีพระคุณทุกๆ ท่าน

จรัสศรี รุ่งรัตนอุบล

มกราคม 2551

|                      |                                                                                  |
|----------------------|----------------------------------------------------------------------------------|
| ชื่อเรื่อง           | การออกแบบการทดลองคอมพิวเตอร์โดยใช้อัลกอริทึมการสืบค้น                            |
| ชื่อเรื่องภาษาอังกฤษ | Design for computer experiment using search algorithms                           |
| หัวหน้าโครงการ       | ดร. จรัสศรี รุ่งรัตนอุบล (70%)<br>ภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ |
| ผู้ร่วมโครงการ       | เกตุจันทร์ จำปาไชยศรี (30%)<br>ภาควิชาคณิตศาสตร์                                 |

งบประมาณวิจัยคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวรประจำปี 2550

### บทคัดย่อ

การจำลองการทำงานด้วยคอมพิวเตอร์นั้นมีความสำคัญอย่างยิ่งในปัจจุบันทั้งนี้อาจเนื่องมาจากกำลังคน เวลา และทรัพยากรที่ต้องใช้ในการทดลองจริงหรือการทดลองทางกายภาพนั้นมีต้นทุนที่สูง ทำให้การจำลองการทำงานด้วยคอมพิวเตอร์เป็นที่นิยม

การสร้างแบบจำลองการทำงานด้วยคอมพิวเตอร์นั้นเป็นอีกหนึ่งงานวิจัยที่ได้รับความนิยมเป็นอย่างยิ่ง ทั้งนี้เพราะต้นทุนหรือเงินที่ต้องใช้ในการจำลองการทำงานด้วยคอมพิวเตอร์ในแต่ละครั้งนั้นมีปริมาณที่สูง และใช้เวลานานในแต่ละครั้งการที่คอมพิวเตอร์ประมวลผล การออกแบบการทดลองที่ดีย่อมส่งผลดีในเรื่องของการลดจำนวนเงิน และเวลาที่ใช้ในการทดลอง ในงานวิจัยนี้ได้นำเทคนิคการสืบค้นมาประยุกต์ใช้เพื่อค้นหาแผนการทดลองที่ดีที่สุด โดยได้นำเทคนิคเจเนติก และซิมูเลตเตด แอนนิลลิง มาใช้ บนคลาสการออกแบบแบบ Latin hypercube designs (LHD) และได้ข้อสรุปว่า ซิมูเลตเตด แอนนิลลิงสามารถหาแผนการทดลองได้ดีกว่าเทคนิคเจเนติกภายใต้เงื่อนไขการสร้างสายพันธุ์และกลายพันธุ์แบบง่าย ภายใต้เกณฑ์วัดค่าที่ดีที่สุด  $\phi_p$

## Abstract

Recent advances in computational power have lead to study of physical process through deterministic computer simulated experiments. Usually these computer simulated experiments are time consuming and computationally expensive. Hence an effort has gone into developing of cheaper and reliable surrogate models that can be replaced the computer simulated experiments. The choice of the design is critical in success of any statistical modeling methods. As the output response from computer simulated experiments are deterministic, therefore the ideas like blocking, replication, and randomization in traditional physical experimentation are irrelevant. Consequently the space filling designs that aim to spread the design points over the design space is desired. Latin hypercube designs (LHD) are widely practiced in the context of computer simulated experiments. In this paper we compare the efficiency of various types of LHD based designs considering prediction accuracy. These comparisons indicate that symmetric LHD designs perform better than other classes for small dimensional problem while the orthogonal LHD designs generated by William transformation are superior over other classes of design for larger dimension of the problem

## สารบัญ

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| ประกาศนุญการ .....                                                           | I   |
| บทคัดย่อ .....                                                               | II  |
| Abstract .....                                                               | III |
| <br>                                                                         |     |
| บทที่ 1 .....                                                                | 1   |
| บทนำ .....                                                                   | 1   |
| ชื่อโครงการ .....                                                            | 1   |
| คณะผู้วิจัย(ระดับสังกัดภาควิชา) และสัดส่วนที่ทำงานวิจัย (%) .....            | 1   |
| สถานที่จัดทำโครงการวิจัย .....                                               | 1   |
| ความสำคัญและที่มาของปัญหาที่ทำการวิจัย .....                                 | 2   |
| กรอบแนวคิดหรือทฤษฎี และงานวิจัยที่เกี่ยวข้อง .....                           | 3   |
| วัตถุประสงค์ .....                                                           | 3   |
| ประโยชน์ที่คาดว่าจะได้รับ .....                                              | 4   |
| ขอบเขตการวิจัย .....                                                         | 4   |
| วิธีดำเนินการวิจัย .....                                                     | 4   |
| ระยะเวลาที่ทำการวิจัย .....                                                  | 5   |
| งบประมาณที่เสนอขอ .....                                                      | 6   |
| <br>                                                                         |     |
| บทที่ 2 .....                                                                | 7   |
| แนวคิดและทฤษฎีที่เกี่ยวข้อง .....                                            | 7   |
| การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment – CSE) .....  | 7   |
| อัลกอริทึมการค้นหา .....                                                     | 16  |
| อัลกอริทึมการสืบค้นแบบเจเนติก – Genetic Algorithm (GA) .....                 | 17  |
| อัลกอริทึมการสืบค้นแบบซิมูเลตเตด แอนนิลลิ่ง – Simulated Annealing (SA) ..... | 21  |
| <br>                                                                         |     |
| บทที่ 3 .....                                                                | 23  |
| วิธีการดำเนินงาน .....                                                       | 23  |
| กำหนดเกณฑ์หาค่าเหมาะสม (Optimality criteria) .....                           | 23  |
| กำหนดค่าพารามิเตอร์ต่างๆ สำหรับอัลกอริทึมการค้นหาทั้ง 2 วิธี .....           | 24  |

|                                |           |
|--------------------------------|-----------|
| ออกแบบการทดลอง.....            | 25        |
| สรุปผลการทดลอง.....            | 25        |
| <b>บทที่ 4 .....</b>           | <b>27</b> |
| <b>สรุปผลการดำเนินงาน.....</b> | <b>27</b> |
| สรุปผลการดำเนินงาน.....        | 27        |
| บรรณานุกรม .....               | 28        |
| ภาคผนวก.....                   | 30        |



## บทที่ 1

### บทนำ

รายงานนี้เป็นส่วนหนึ่งของโครงงานวิจัยของคณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร ประจำปีงบประมาณ 2550 โดยผู้จัดทำได้รับการอนุมัติทำงานชิ้นนี้ โดยมีชื่อโครงการและรายละเอียดโครงการดังต่อไปนี้

#### ชื่อโครงการ

(ภาษาไทย) การออกแบบการทดลองคอมพิวเตอร์โดยใช้อัลกอริทึมการสืบค้น

(ภาษาอังกฤษ) Design for computer experiment using search algorithms

#### คณะผู้วิจัย(ระบุสังกัดภาควิชา) และสัดส่วนที่ทำงานวิจัย (%)

##### หัวหน้าโครงการวิจัย

ดร. จรัสศรี รุ่งรัตนอุบล (70%)

Ph.D. Computer Science

พนักงานสายวิชาการ สังกัดภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

##### ผู้ร่วมโครงการวิจัย

ดร. เกตุจันทร์ จำปาไชยศรี (30%)

Ph.D. Statistics

อาจารย์ สังกัดภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์  
มหาวิทยาลัยนเรศวร

#### สถานที่จัดทำโครงการวิจัย

ภาควิชาคณิตศาสตร์และภาควิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะ  
วิทยาศาสตร์ มหาวิทยาลัยนเรศวร

## ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

การศึกษาวิจัยในปัจจุบันได้มีการนำเทคนิคการวางแผนทดลองเข้ามาช่วยในการเก็บรวบรวมข้อมูลเพื่อทำการวิจัยในขั้นสูงต่อไป หลักการวางแผนการทดลองโดยทั่วไปนั้นสามารถทำได้โดยการกำหนดค่าของตัวแปรอิสระ (Input Variable) และทำการวางแผนทดลองเพื่อเก็บข้อมูลของค่าตัวแปรตาม (Output Response) ในกรณีที่ผู้ศึกษาต้องการทำการทดลองเพื่อศึกษาเกี่ยวกับระบบที่มีความซับซ้อน ซึ่งไม่สามารถทำการทดลองทางกายภาพ (Physical Experiments) ได้ ยกตัวอย่างเช่น การพยากรณ์ปริมาณน้ำมันใต้ดินในบางพื้นที่เมื่อทราบคุณลักษณะของตัวแปรอิสระที่เกี่ยวข้องในพื้นที่นั้น ๆ การศึกษาลักษณะการไหลเวียนของเลือดเมื่อทดลองฉีดยาให้กับผู้ป่วย จะเห็นได้ว่าการทดลองลักษณะนี้ ไม่สามารถทำการทดลองกับหน่วยทดลองจริง ๆ ได้ ฉะนั้นจึงต้องมีการทำการทดลองหรือการจำลองอีกแบบหนึ่งขึ้นมาซึ่งมีชื่อเรียกว่า การทดลองด้วยคอมพิวเตอร์ (Computer Experiment) ซึ่งการทดลองในลักษณะนี้สามารถทำได้โดยใช้เทคนิคการจำลองทางคอมพิวเตอร์ทำการประมวลผลค่าของระดับต่าง ๆ ของตัวแปรอิสระหรือแต่ละหน่วยทดลองโดยใช้เทคนิค numerical method เช่น finite element code ทำการประมวลผลเพื่อให้ได้ค่าของตัวแปรตาม จากนั้นก็นำค่าตัวแปรตามที่ได้นี้ไปใช้ทำการวิเคราะห์เพื่อการพยากรณ์หรือเพื่อสร้างตัวแบบต่อไป

โดยทั่วไปการทำการทดลองด้วยคอมพิวเตอร์นั้น จะสิ้นเปลืองเวลาและทรัพยากรมาก เช่น ในการใช้ finite element code เพื่อทำการประมวลผลอาจต้องใช้เวลาเป็นชั่วโมงหรือวันเพื่อให้ได้ค่าของตัวแปรตามจากแต่ละหน่วยทดลอง ซึ่งจะเห็นได้ว่าถ้าการทดลองประกอบไปด้วยจำนวนหน่วยทดลองที่มีจำนวนมาก การประมวลผลจาก finite element code เพื่อให้ได้ค่าตัวแปรตามครบตามต้องการก็จะใช้เวลามากตามไปด้วย ฉะนั้นจึงได้มีการคิดค้นเทคนิคต่าง ๆ เข้ามาช่วยเพื่อลดค่าใช้จ่ายและเวลาจากการทำการทดลองด้วยคอมพิวเตอร์ ซึ่งเทคนิคที่ถูกพัฒนาอย่างแพร่หลายคือ การสร้างตัวแบบเพื่อให้อธิบายความสัมพันธ์ของตัวแปรอิสระและตัวแปรตามโดยใช้จำนวนหน่วยทดลอง ( $n$ ) น้อยที่สุดเท่าที่จะเป็นไปได้ และการใช้ตัวแบบเพื่อการพยากรณ์เมื่อทราบค่าระดับต่าง ๆ ของตัวแปรอิสระที่อยู่ในบริเวณที่เราสนใจศึกษา ในการพัฒนาตัวแบบที่มีความน่าเชื่อถือนั้น จะต้องมีการวางแผนทดลองที่ดีเพื่อให้ได้ค่าของตัวแปรตาม ที่สามารถใช้แสดงถึงความสัมพันธ์ระหว่างตัวแปรอิสระและตัวแปรตามที่แท้จริงให้มากที่สุด ซึ่งหลักการวางแผนทดลองสำหรับคอมพิวเตอร์นั้นจะต้องทำการกระจายหน่วยทดลองให้มากที่สุด และเป็นตัวแทนของบริเวณที่เราสนใจศึกษาให้มากที่สุดนั่นเอง

เนื่องจากการทดลองด้วยคอมพิวเตอร์ มักจะประกอบไปด้วยจำนวนตัวแปรอิสระที่ต้องการศึกษาจำนวนหลายตัว และตัวแปรอิสระแต่ละตัวประกอบไปด้วยระดับค่าต่าง ๆ หลาย

ระดับ ฉะนั้นเทคนิคการออกแบบการทดลองแบบคลาสสิก (Classical Experiment) จึงไม่เหมาะสมที่จะนำมาใช้กับการทดลองแบบนี้ได้ นักวิจัยส่วนใหญ่จึงได้มีการคิดค้นการวางแผนทดลองโดยใช้ความสามารถของคอมพิวเตอร์เพื่อช่วยให้สามารถสร้างแผนการทดลองได้อย่างมีประสิทธิภาพ ซึ่งเทคนิคนี้มีชื่อเรียกว่า อัลกอริทึมการสืบค้น (Search algorithm) ในการใช้ อัลกอริทึมการสืบค้นนั้นจะต้องมีการกำหนดเกณฑ์ในการเลือกค่าที่เหมาะสม (Optimality criteria) เข้ามาช่วยในการตัดสินใจว่าจะหยุดการสืบค้นและรายงานแผนการทดลองที่ดีที่สุดเมื่อใด ซึ่งเกณฑ์ในการเลือกค่าที่เหมาะสมที่ถูกนำมาใช้อย่างแพร่หลายคือ เกณฑ์ค่าต่ำสุดที่มากที่สุด (Maximin) และเกณฑ์ไฟล์ ( $\phi_p$ ) โดยที่เกณฑ์ทั้งสองนี้สามารถหาค่าได้จากการคำนวณระยะห่างระหว่าง 2 หน่วยทดลองใด ๆ นั่นเอง

โดยทั่วไปการใช้ อัลกอริทึมการสืบค้นควบคู่กับเกณฑ์ในการเลือกค่าที่เหมาะสมเพื่อค้นหาแผนการทดลองที่เหมาะสมในแต่ละมิติปัญหานั้น จะใช้เวลานานมาก โดยเฉพาะอย่างยิ่งในกรณีที่มีจำนวนตัวแปรอิสระที่ต้องการศึกษา (d) และจำนวนรัน (n) มีจำนวนมาก ฉะนั้นงานวิจัยนี้จึงมีวัตถุประสงค์ที่จะค้นหาแผนการทดลองที่ดีที่สุดโดยใช้ อัลกอริทึมการสืบค้นให้เกิดประโยชน์สูงสุดและประหยัดเวลาในการค้นหาให้มากที่สุดเท่าที่จะเป็นไปได้ รวมไปถึงการกำหนดค่าพารามิเตอร์ตั้งต้นที่เหมาะสมที่สุดเพื่อใช้สำหรับแต่ละอัลกอริทึมการสืบค้น

#### กรอบแนวคิดหรือทฤษฎี และงานวิจัยที่เกี่ยวข้อง

หลักการพัฒนาอัลกอริทึมการสืบค้นบนพื้นฐานของค่าพารามิเตอร์ตั้งต้นที่เหมาะสมที่สุดจะถูกศึกษาในงานวิจัยนี้ โดยจะใช้เกณฑ์ของการเลือกค่าที่เหมาะสมเป็นมาตรฐานในการวัดประสิทธิภาพของแต่ละอัลกอริทึมการสืบค้นซึ่งจะถูกวัดโดยใช้ค่าอัตราความเร็วของการลู่เข้าสู่ค่าที่เหมาะสม (Rate of Convergence) และความเรียบง่ายของอัลกอริทึม (Simplicity) ภายใต้กรอบเวลาที่เท่ากัน จากนั้นค่าข้อมูลที่ได้จากแต่ละอัลกอริทึมการสืบค้นซึ่งถูกบันทึกจากมิติปัญหาที่แตกต่างกัน จะถูกนำไปทดสอบความแตกต่างโดยใช้วิธีวิเคราะห์ทางสถิติ เช่น Analysis of Variance (ANOVA) และ T-test เป็นต้น

#### วัตถุประสงค์

- 1) ศึกษาพารามิเตอร์ตั้งต้นที่เหมาะสมที่สุดสำหรับอัลกอริทึมการสืบค้น
- 2) เพื่อเปรียบเทียบประสิทธิภาพของอัลกอริทึมการสืบค้นแต่ละประเภทโดยใช้ตัวแทนที่ดีที่สุดจากแต่ละอัลกอริทึมการสืบค้น

### ประโยชน์ที่คาดว่าจะได้รับ

ผลจากงานวิจัยในเรื่องนี้จะเป็นแนวทางในการเลือกใช้อัลกอริทึมเพื่อการค้นหาแผนการทดลองที่เหมาะสม เพื่อนำไปใช้สำหรับการทดลองทางคอมพิวเตอร์ นอกจากนี้ผู้วิจัยยังสามารถนำหลักการจากงานวิจัยนี้เพื่อนำไปพัฒนาอัลกอริทึมเพื่อการค้นหาแบบอื่นๆ รวมไปถึงการกำหนดค่าต่าง ๆ ของพารามิเตอร์ตั้งต้นที่เหมาะสมได้

### ขอบเขตการวิจัย

งานวิจัยนี้ประกอบด้วยการเปรียบเทียบประสิทธิภาพของอัลกอริทึมเพื่อการค้นหาภายใต้เงื่อนไขในการเลือกค่าที่เหมาะสมโดยจะทำการทดลองกับมิติของปัญหาต่าง ๆ ดังรายละเอียดต่อไปนี้

#### ประเภทของอัลกอริทึม

- 1) Simulated Annealing Algorithm (SA)
- 2) Genetic Algorithm (GA)

#### ประเภทของเกณฑ์ในการเลือกค่าที่เหมาะสม

- 1) Maximin
- 2)  $\phi_p$
- 3) มิติของปัญหาที่จะทดสอบ

| d | 2 | 3  | 4  | 5  | 6  | 7  | 8   | 9   | 10  |
|---|---|----|----|----|----|----|-----|-----|-----|
| g | 3 | 4  | 5  | 6  | 7  | 8  | 9   | 10  | 11  |
|   | 4 | 7  | 12 | 17 | 23 | 30 | 39  | 48  | 58  |
|   | 6 | 12 | 19 | 29 | 40 | 54 | 69  | 87  | 106 |
|   | 8 | 16 | 27 | 40 | 57 | 77 | 100 | 125 | 154 |
|   | 9 | 19 | 33 | 51 | 73 | 99 | 129 | 163 | 201 |

### วิธีดำเนินการวิจัย

- 1) ศึกษาลักษณะการทำงานของอัลกอริทึมเพื่อการสืบค้นหารวมไปถึงเกณฑ์ในการเลือกค่าที่เหมาะสม
- 2) กำหนดมิติของปัญหาที่ต้องการศึกษา
- 3) พัฒนาโปรแกรม

- 4) จำลองการทำงานของโปรแกรมเพื่อหาพารามิเตอร์ที่เหมาะสมที่สุดของแต่ละอัลกอริทึมการสืบค้น
- 5) รันโปรแกรมเพื่อเก็บรวบรวมข้อมูล
- 6) วิเคราะห์ข้อมูล
- 7) สรุปผลการดำเนินการพัฒนาอัลกอริทึมเพื่อการค้นหา และจัดทำรูปแบบรายงาน

#### ระยะเวลาที่ทำการวิจัย

งบประมาณปี 2550 ตั้งแต่วันที่ 1 ตุลาคม 2549 ถึง 30 กันยายน 2550

#### แผนการดำเนินการตลอดโครงการ

| กิจกรรม                                                                                                                                         | เดือนที่ |   |   |   |   |   |   |   |   |    |    |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------|---|---|---|---|---|---|---|---|----|----|----|
|                                                                                                                                                 | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| ศึกษาลักษณะ โครงสร้าง รูปแบบ ลักษณะการทำงานของอัลกอริทึมเพื่อการค้นหา ศึกษาปัญหาและอุปสรรคต่าง ๆ ข้อดีและข้อเสียของแต่ละอัลกอริทึมเพื่อการค้นหา | ↔        |   |   |   |   |   |   |   |   |    |    |    |
| ศึกษาการกำหนดพารามิเตอร์เบื้องต้นที่เกี่ยวข้องในแต่ละอัลกอริทึมเพื่อการค้นหา                                                                    |          | ↔ |   |   |   |   |   |   |   |    |    |    |
| ออกแบบอัลกอริทึมและเขียนเพื่อจำลองบนโปรแกรม R                                                                                                   |          |   | ↔ |   |   |   |   |   |   |    |    |    |
| จำลองโปรแกรมและบันทึกข้อมูลของแต่ละมิติปัญหา                                                                                                    |          |   |   |   | ↔ |   |   |   |   |    |    |    |
| วิเคราะห์ข้อมูลและหาข้อสรุป                                                                                                                     |          |   |   |   |   |   |   | ↔ |   |    |    |    |
| จัดทำรูปแบบฉบับสมบูรณ์                                                                                                                          |          |   |   |   |   |   |   |   |   | ↔  |    |    |

#### เอกสารอ้างอิง

- Morris, M.D., Mitchell, T.J., 1995. Exploratory design for computer experiments. Journal of Statistical planning and inference 43, 381-402.
- Booker, A.J., 1998. Design and analysis of computer experiments. AIAA-98-4757. American Institute of Aeronautics and Astronautics.
- Li, Z, Shigeru, N., 2001. Maximin distance-lattice hypercube design for computer experiment based on genetics algorithm. Proceedings. ICII 2001

Beijing. 2001 International Conferences, Beijing China, 29 October 2001 – 1 November 2001, vol. 2, pp. 814-819.

Montgomery, D.C., 2001. Design and analysis of Experiments. John Wiley&Sons, New York.

ดร. จรัญ จันทลักษณ์. สถิติ วิธีวิเคราะห์และวางแผนงานวิจัย, สำนักพิมพ์ ไทยวัฒนาพานิช จำกัด, กรุงเทพฯ

### งบประมาณที่เสนอขอ

| รายการ                                            | งบประมาณที่เสนอขอ (บาท) |
|---------------------------------------------------|-------------------------|
| <b>1. หมวดค่าตอบแทน</b>                           |                         |
| - ค่าตอบแทนนิสิตช่วยงาน (2 คน x 200 บาท x 50 วัน) | 20,000                  |
| - ค่าตอบแทนการปฏิบัติงานนอกเวลาราชการ             | 10,000                  |
| - ค่าตอบแทนที่ปรึกษาโครงการ                       | 1,000                   |
| - ค่าตอบแทนผู้เชี่ยวชาญตรวจสอบเครื่องมือวิจัย     | 1,000                   |
| <b>2. หมวดวัสดุ</b>                               |                         |
| - ค่าหนังสือตำรา และวารสาร                        | 4,000                   |
| - ค่าวัสดุสำนักงาน                                | 1,500                   |
| - ค่าวัสดุคอมพิวเตอร์                             | 8,000                   |
| <b>3. หมวดค่าใช้สอย</b>                           |                         |
| - ค่าจ้างวิเคราะห์ข้อมูล                          | 1,500                   |
| - ค่าถ่ายเอกสาร ค่าเช่าเล่ม ค่าพิมพ์รายงาน        | 3,000                   |
| <b>รวมงบประมาณที่เสนอขอ</b>                       | <b>50,000</b>           |

(ทั้งนี้สามารถถัวเฉลี่ยทุกรายการ)

## บทที่ 2

### แนวคิดและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่ใช้ในการดำเนินการวิจัยโดยในที่นี้สามารถแจกแจงได้ดังต่อไปนี้

- 1) การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment – CSE)
- 2) อัลกอริทึมการค้นหา (Search Algorithms)

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

การจำลองการทดลองด้วยคอมพิวเตอร์ (Computer Simulated Experiment: CSE) ได้ถูกนำมาใช้อย่างแพร่หลายในงานด้านวิทยาศาสตร์และวิศวกรรมศาสตร์ โดยเฉพาะอย่างยิ่งในการออกแบบระบบที่มีความซับซ้อน ที่การทดลองจริงทางกายภาพเป็นไปได้ยากหรือเป็นไปได้ไม่ เช่น อากาศยานและระบบการขนส่ง (aircraft and logistics system), การพยากรณ์อากาศ, การศึกษาพฤติกรรมของโครงสร้าง (ตึกและชิ้นส่วนอุตสาหกรรม) ภายใต้ภาวะกดดันและเงื่อนไขต่างๆ, เครื่องจำลองปฏิกิริยาสำหรับวิศวกรรมปิโตรเคมี, การจำลองการทำปฏิกิริยาทางเคมี รวมไปถึงการทำงานของนิวเคลียร์ เป็นต้น เพราะมีข้อจำกัดต่างๆ เช่น งบประมาณ, ผลกระทบต่อสิ่งแวดล้อม, ข้อจำกัดทางด้านกฎหมาย, เทคโนโลยีไม่สนับสนุน เป็นต้น

ประโยชน์ของการจำลองการทดลองด้วยคอมพิวเตอร์ มีด้วยกันมากมายหลายประการ ดังตัวอย่างต่อไปนี้

(1) ลดการสูญเสียทรัพยากร เช่น การสร้างชิ้นส่วนปีกเครื่องบิน การซ่อมรถ การทดลอง ยา ถ้าใช้การจำลองเหตุการณ์ ไม่ต้องเสียค่าวัสดุอุปกรณ์ ไม่ต้องเสียทหาร ไม่ต้องเสียชีวิตคนไข้

(2) ประหยัดเวลา ประหยัดทรัพยากร เพราะทดลองในห้องทดลอง บนโต๊ะทำงาน หรือทดลองกับคอมพิวเตอร์ ค่าใช้จ่ายจึงน้อย การแก้สมการต่างๆ บางครั้งการทดลองแทนค่าจะเร็วกว่าการแก้สมการตามปกติ

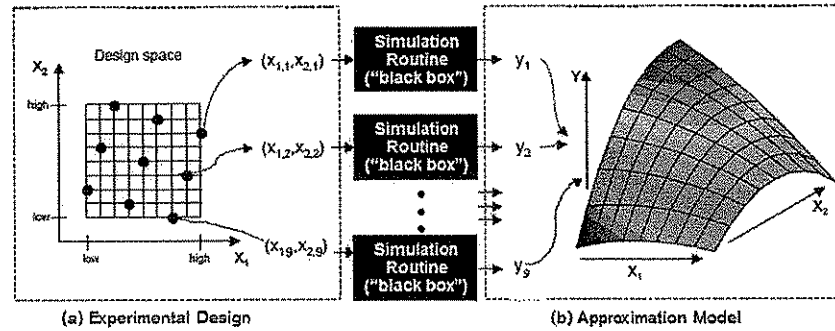
(3) หากสร้างแบบจำลองที่ดี และใช้คอมพิวเตอร์ช่วย เราจะเห็นภาพหรือตัวเลขการเปลี่ยนแปลงของตัวแปรต่างๆในระบบเป็นขั้นเป็นตอน ดีกว่าการหาคำตอบสุดท้ายธรรมดา (ผศ. ดร.บุญเสริม บุญเจริญผล)

การจำลองการทดลองด้วยคอมพิวเตอร์ มีการนิยามไว้หลายความหมาย คือ เป็นเทคนิคกระทำการทดลองกับแบบจำลองของระบบ การทดลองกระทำกับแบบจำลอง ไม่ต้องกระทำกับของจริง (Hillier and Lieberman, 1980), เป็นการสร้างเครื่องมือทดลอง ซึ่งจะทำงานเลียนแบบระบบของเรื่องที่น่าสนใจได้อย่างรวดเร็ว และประหยัด (Eppen, Gould and Schmidt, 1985), เป็นเทคนิคเพื่อการพัฒนา ทดสอบ และประเมินผล โดยสมมติเหตุการณ์แวดล้อมเสมือนว่าเป็นจริงก่อนที่จะลงมือปฏิบัติการจริง เป็นวิธีการที่ใส่ค่าตัวแปรอิสระ (Input variable) เข้าไปในสมการและจะได้ค่าของตัวแปรตาม (Output response) หรือค่าคำตอบ กระบวนการเหล่านี้เป็นการสมมติเหตุการณ์เสมือนจริงหรือรู้สึกเหมือนจริง โดยมีเป้าหมายแน่นอน ทำให้ผู้ใช้ได้เรียนรู้โดยการกระทำ ได้ผลที่เป็นจริงโดยไม่ต้องเกิดการสูญเสีย ซึ่งจะเกิดผลสัมฤทธิ์สูงสุดได้ก็ต่อเมื่อมีการออกแบบระบบอย่างละเอียดรอบคอบ (ผศ.ดร.บุญเสริม บุญเจริญผล, 2547)

ในงานวิจัยนี้ การจำลองการทดลองด้วยคอมพิวเตอร์ หมายถึง ใช้โปรแกรมคอมพิวเตอร์เพื่อจำลองรูปแบบของระบบใดๆ โดยอาศัยข้อมูลต่างๆ ที่เกี่ยวข้อง (Input variable) ร่วมกับสมการทางคณิตศาสตร์เพื่ออธิบายการทำงานของระบบและได้ค่าของตัวแปรตาม (Output variable) เพื่อใช้เป็นตัวแทนของการทำการทดลองทางกายภาพ

Simpson et al. (2000) ได้อธิบายกระบวนการในการจำลองการทดลองด้วยคอมพิวเตอร์ว่ามีส่วนประกอบด้วยกันทั้งหมด 3 ส่วน คือ

- 1) Experimental design คือ การวางแผนการทดลอง ซึ่งประกอบด้วยค่าของปัจจัยที่เกี่ยวข้องระดับต่างๆ ของตัวแปรเข้า (Treatment combination)
- 2) Simulation Routine หรือ กล่องดำ (black box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดการณ์ได้ว่าจะเกิดอะไรขึ้นภายใน ซึ่งกระบวนการใน black box นี้จะใช้รูปแบบสมการทางคณิตศาสตร์ที่มีความซับซ้อน เพื่อให้ได้ผลลัพธ์ออกมา ซึ่งผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ ของปัจจัยที่เกี่ยวข้องที่เราป้อนเข้าไป
- 3) Approximation Model คือ การสร้างโมเดลเพื่อการประมาณ เกิดจากการนำแผนการทดลองกับผลลัพธ์ที่ได้หลังจากกระบวนการในกล่องดำ มาสร้างรูปแบบความสัมพันธ์ที่เหมาะสมให้สามารถนำไปใช้ประโยชน์ต่อไป ส่วนประกอบต่างๆ เหล่านี้สามารถแสดงได้ด้วยรูปที่ 1



รูปที่ 1 แสดงส่วนประกอบหลักของ CSE ดัดแปลงจาก (Simpson, Lin and Chen.W, 2000)

จากรูปที่ 1(a) แสดงกลุ่มของค่าระดับต่างๆ ของปัจจัยที่เกี่ยวข้องที่เราสนใจศึกษา (design space) ที่เลือกมาเป็นแผนการทดลอง โดยทั่วไปจะถูกกำหนดด้วยค่าขอบเขตค่ามากที่สุดและค่าน้อยสุดของแต่ละตัวแปร (design input) จากรูปที่ 1  $X_1$  และ  $X_2$  เป็นตัวอย่างตัวแปรที่สนใจศึกษา ในการทดลองหนึ่งๆ ผู้ออกแบบจะต้องวางแผนการทดลองโดยเลือกแผนการทดลองมาหนึ่งชุดจาก design space โดยต้องเลือกค่าของแต่ละตัวแปรให้ครอบคลุมพื้นที่ใน design space ให้ได้มากที่สุดโดยมีเงื่อนไขว่าต้องใช้จำนวนจุด (Run) ให้น้อยที่สุดเพื่อเป็นการประหยัดเวลาและทรัพยากรในการจำลองการทดลองด้วยคอมพิวเตอร์

ในการเลือกแผนการทดลองนั้น ถ้าทำการเลือกโดยที่ไม่มีการกำหนดช่วงหรือระยะห่าง (interval) ระหว่างค่าสูงสุดกับต่ำสุด จะทำให้จำนวนทางเลือกที่เป็นไปได้มากมายไร้ขีดจำกัด หรือมีทางเลือกเป็นจำนวนอนันต์ (Infinite number) เช่น ถ้าเลือกค่าระหว่าง 1 ถึง 3 จะเห็นได้ว่า ค่าที่เลือกอาจจะเป็นค่าใดๆ ก็ได้ และจะเกิดทางเลือกแบบอนันต์ เช่น 1, 1.131, 1.0423, 2.56, 2.031234, ..., 3 เป็นต้น ดังนั้นจึงจำเป็นต้องกำหนดช่วงหรือระยะห่างของ design space ขึ้น เพื่อให้การเลือกกลุ่มของค่าของแต่ละตัวแปร (Treatment combination) ที่เหมาะสม

การแบ่งช่วงของ design space ในการทดลองด้วยคอมพิวเตอร์นั้นมีด้วยกันหลายประเภท หรือหลายคลาส ซึ่งกลุ่มการออกแบบแผนการทดลองนั้นจะถูกเรียกว่าคลาสของแผนการทดลอง (Class of design) ตัวอย่างของคลาสของแผนการทดลองที่ใช้กันในปัจจุบัน ได้แก่ Latin Hypercube design (LHD), Hammersley Sequence Sampling, Orthogonal Arrays, Uniform Design (Giunta, Wojtkiewicz Jr and Eldred, 2003) เป็นต้น

ในที่นี้ขอยกตัวอย่างลักษณะการกำหนดช่วงผ่านรูปที่ 1(a) โดยกำหนดช่วงให้กับแต่ละตัวแปรเป็น 9 ช่วงเท่าๆ กัน โดยในแผนทดลองนี้จะทำการทดลองทั้งหมด 9 รัน (Run) คำว่า รัน (Run) ในที่นี้ คือจำนวนครั้งของการทดลองในแผนทดลองนี้ ตัวแปรเข้า (Input variable) ในการทดลองนี้ มีทั้งหมด 2 ตัว คือ  $X_1$  และ  $X_2$  ซึ่งหนึ่งแผนทดลองจะประกอบด้วย คู่อันดับ ( $X_{1,i}, X_{2,i}$ ) จำนวน 9 คู่ อันดับ ค่าของ  $X_{1,1}, X_{1,2}, X_{1,3}, \dots, X_{1,9}$  จะอยู่ในช่วง [Lower bound, Upper

bound] ของ  $X_1$  และค่าของ  $X_{2,1}, X_{2,2}, X_{2,3}, \dots, X_{2,9}$  จะอยู่ในช่วง [Lower bound, Upper bound] ของ  $X_2$  จากกรณีนี้ จะสามารถเขียนเมตริกซ์ของการวางแผนการทดลองที่มีมิติ  $9 \times 2$  ได้ดังรูปที่ 2

$$\begin{pmatrix} X_{1,1} & X_{2,1} \\ X_{1,2} & X_{2,2} \\ X_{1,3} & X_{2,3} \\ X_{1,4} & X_{2,4} \\ X_{1,5} & X_{2,5} \\ X_{1,6} & X_{2,6} \\ X_{1,7} & X_{2,7} \\ X_{1,8} & X_{2,8} \\ X_{1,9} & X_{2,9} \end{pmatrix}$$

รูปที่ 2 เมตริกซ์แผนการทดลองมิติ  $9 \times 2$  ของตัวแปร  $X_1, X_2$

จากแผนการทดลองมิติ  $9 \times 2$  ในรูปที่ 2 เมื่อนำมาใช้กับคลาส LHD ซึ่งเป็นที่นิยมใช้กันในการวางแผนการทดลอง ลักษณะแผนการทดลองที่เป็นไปได้ มีได้มากมายหลายแบบ ดังแสดงในรูปที่ 3

$$(9!)^2 = 1.3168 \times 10^{11} \text{ แบบ}$$

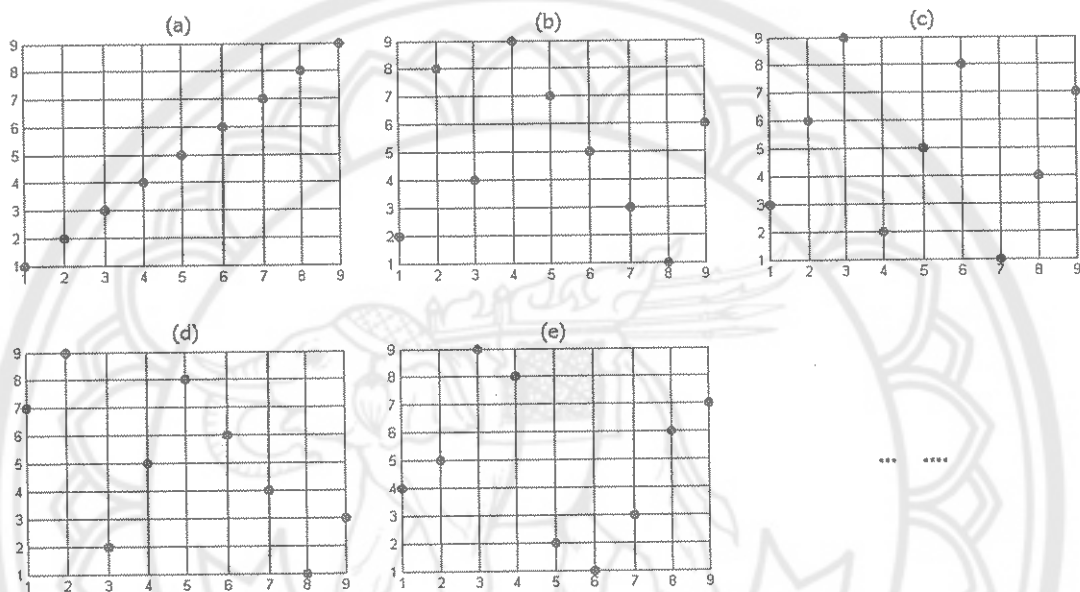
|     |     |     |     |            |
|-----|-----|-----|-----|------------|
| 1 1 | 2 8 | 1 3 | 1 7 | 6 1        |
| 2 2 | 3 4 | 2 6 | 2 9 | 3 9        |
| 3 3 | 4 9 | 3 9 | 3 2 | 8 6        |
| 4 4 | 7 3 | 4 2 | 5 8 | 7 3        |
| 5 5 | 5 7 | 5 5 | 7 4 | 1 4 ... .. |
| 6 6 | 1 2 | 6 8 | 4 5 | 5 2        |
| 7 7 | 6 5 | 7 1 | 9 3 | 4 8        |
| 8 8 | 9 6 | 8 4 | 6 6 | 2 5        |
| 9 9 | 8 1 | 9 7 | 8 1 | 9 7        |
| (a) | (b) | (c) | (d) | (e)        |

รูปที่ 3 ตัวอย่างแผนการทดลองมิติ  $9 \times 2$  ที่เป็นไปได้

จากรูปที่ 3 เป็นตัวอย่างแผนการทดลองมิติ  $9 \times 2$  ที่เป็นไปได้บางส่วน และเป็นไปตามลักษณะของ LHD คือ ในแต่ละหลักต้องมีค่าไม่ซ้ำกันเป็นค่าในแต่ละช่วง ใช้ตัวเลข 1 – 9 เป็นตำแหน่งแทนค่าที่อยู่ในตำแหน่งนั้น ทำให้ในแต่ละหลักสามารถที่จะเรียงค่าได้เป็น  $9!$  จาก

ตัวอย่างแผนการทดลองมิติ  $9 \times 2$  ให้แผนการทดลองทั้งหมดที่เป็นไปได้ เท่ากับ  $9! \times 9!$  หรือ  $(9!)^2 = 362880^2 = 1.3168 \times 10^{11}$  ดังรูปที่ 3

ถ้าจำนวนตัวแปรเข้าและจำนวนการทดลองมีค่าเพิ่มขึ้น เมตริกซ์แผนการทดลองที่เป็นไปได้ จะเพิ่มขึ้นแบบทวีคูณ ถ้ากำหนดให้  $d$  หมายถึงจำนวนตัวแปรเข้าที่เกี่ยวข้องกับการทดลองนี้ และ  $n$  เป็นจำนวนการทดลองหรือจำนวนวัน จะได้เมตริกซ์แผนการทดลองเป็นมิติ  $n \times d$  และจะได้จำนวนแผนการทดลองที่เป็นไปได้ทั้งหมดเท่ากับ  $(n!)^d$



รูปที่ 4 ตัวอย่างแผนการทดลองบนระนาบ 2 มิติ

จะเห็นว่าแผนการทดลองที่เป็นไปได้มีจำนวนมหาศาล ถ้านำแผนการทดลองที่ได้แต่ละแผนมากำหนดจุดบนระนาบ ดังรูปที่ 4 จะเห็นว่าแผนการทดลองที่ดีควรจะครอบคลุมพื้นที่ design space ให้ได้มากที่สุด แผนการทดลองที่ครอบคลุมพื้นที่ได้มากจัดเป็นแผนการทดลองที่ดี (Space filling design) ดังแสดงดังรูปที่ 4 จะเห็นว่ารูปที่ 4(c) เป็นแผนการทดลองที่ดี เพราะครอบคลุมพื้นที่ได้ทั่วทั้ง design space แผนการทดลองที่มีตัวแปร 2 ตัว นั้นเราสามารถตรวจสอบลักษณะการกระจายจุดทดลองของแผนการทดลองได้ง่ายบนระนาบ 2 มิติ แต่ถ้าจำนวนตัวแปรเข้า (d) มีมากขึ้นจะทำให้มิติมากขึ้น ทำให้การกำหนดจุดบนระนาบ d มิติ ทำความเข้าใจได้ยากและเป็นเหตุให้ไม่สามารถเปรียบเทียบแผนการทดลองได้ด้วยการวาดจุดบนระนาบ ดังนั้นการใช้เกณฑ์หาค่าที่เหมาะสม (Optimality criteria) จึงถูกนำมาใช้พิจารณาในการเลือกแผนการทดลองที่ดีที่สุดในแต่ละคลาส เกณฑ์ของค่าที่เหมาะสมที่ถูกนำมาใช้กันอย่างแพร่หลาย โดยทั่วไปเกณฑ์เหล่านี้ถูกนำมาใช้ควบคู่กับอัลกอริทึมการสืบค้น เพื่อช่วยค้นหา

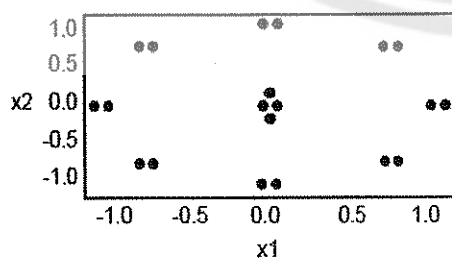
แผนการทดลองที่ดีที่สุดในแต่ละมิติปัญหาที่ถูกกำหนดไว้ เช่น Maximin distance criterion and  $\phi_p$  criterion, Entropy criterion, Centered  $L_2$  discrepancy criterion (Jin, Chen and Sudjianto, 2004)

### การวางแผนการทดลอง (Experimental design)

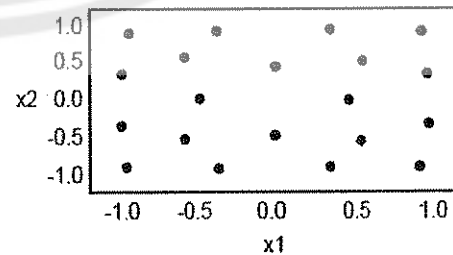
การวางแผนการทดลองมีความสำคัญอย่างยิ่งต่อการสร้างโมเดลที่มีประสิทธิภาพ เพื่อให้ได้โมเดลต้นแบบที่ทำงานได้ใกล้เคียงกับระบบ (Prototype) อย่างถูกต้องแม่นยำที่สุด (Montgomery, 1992)

หลักการวางแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์สามารถแบ่งเป็น 2 แบบ คือ แบบ non-search based (Classical Design) และแบบ search based

1) non-search based (Classical Design) เป็นการวางแผนการทดลองที่เลียนแบบการทดลองทางกายภาพ เช่น central composite, factorial designs เป็นต้น แผนการทดลองในลักษณะนี้จะเป็นแบบที่ไม่แน่นอน (non-deterministic) ขึ้นกับปัจจัยแวดล้อมอื่นๆ ที่อาจส่งผลกระทบต่อผลลัพธ์ที่ได้ในแต่ละครั้งของการทดลอง โดยเทคนิคที่ใช้จะเป็นการสุ่มค่าที่แตกต่างกันเพื่อกระจายค่าให้ครอบคลุม design space แล้วซ้ำค่าของจุดทดลอง (Replicates) แสดงดังรูปที่ 5(a) และ Sack, et al. (1989) ได้เสนอว่าแนวคิดของการวางแผนการทดลองแบบคลาสสิก (Classical Design) ที่เหมาะสมจะเป็นการใช้การรวมกลุ่ม (Blocking) การทำซ้ำ (Replication) และการสุ่ม (Randomization) ซึ่งธรรมชาติของแผนการทดลองลักษณะนี้จะแตกต่างจากการจำลองการทดลองด้วยคอมพิวเตอร์ (CSE) เพราะการทำงานของงานการจำลองการทดลองด้วยคอมพิวเตอร์ (CSE) นั้นจะเป็นแบบแน่นอน (Deterministic) กล่าวคือ เมื่อใส่ค่าตัวแปรเข้าเหมือนกันในทุกๆ ครั้งผลลัพธ์ที่ได้ก็ต้องเหมือนกันในทุกๆ ครั้งด้วย ดังนั้นการวางแผนการทดลองสำหรับการจำลองด้วยคอมพิวเตอร์จึงเป็นแบบ space filling กล่าวคือ พยายามที่จะหาจุดทดลองที่ครอบคลุมเพื่อที่ design space ให้ได้มากที่สุด



(a) "Classical" Design



(b) "Space Filling" Design

รูปที่ 5 การวางแผนการทดลองทั้งสองแบบ

2) Search based เป็นการวางแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ (CSE) โดยจะเน้นที่การเลือกค่าให้กระจายให้ทั่ว design space ให้มากที่สุดเท่าที่จะเป็นไปได้ และไม่จำเป็นต้องทำซ้ำ เพราะผลลัพธ์ (Output) ที่ได้จากการจำลองการทดลองด้วยคอมพิวเตอร์ (CSE) เป็นรูปแบบที่แน่นอน (Deterministic) ดังรูปที่ 5 (b)

การวางแผนการทดลองแบบ Space filling มีความเหมาะสมกับแผนการทดลองสำหรับการจำลองการทดลองด้วยคอมพิวเตอร์ (CSE) อย่างมาก โดยการทำงานจะเป็นการค้นหาจุดทดสอบให้ครอบคลุมทั้ง design space ให้มากที่สุด ในที่นี้การนำอัลกอริทึมการสืบค้นมาประยุกต์ใช้เพื่อค้นหาแผนการทดลองแบบ Search based จึงมีความเหมาะสมเป็นอย่างยิ่ง การนำอัลกอริทึมการสืบค้นเข้ามาใช้เลือกแผนการทดลองที่ดีที่สุด กล่าวคือเลือกค่าจุดที่ครอบคลุม design space ให้ได้ทั้งหมด ดังรูป 5(b) โดยการทำงานดังกล่าวนี้จะต้องทำงานควบคู่ไปกับเกณฑ์หาค่าเหมาะสม (Optimality criteria) นอกจากนี้ปัจจัยที่สำคัญอีกประการหนึ่งคือการแบ่งช่วงของค่าของตัวแปรเข้า ในที่นี้เราเรียกว่าคลาสของแผนการทดลอง

#### คลาสของแผนการทดลอง (Class of Design)

รูปแบบของการวางแผนการทดลองแบบ Space filling ของการจำลองการทดลองด้วยคอมพิวเตอร์นั้นมีหลายแบบด้วยกัน โดยแต่ละแบบจะมีหลักการและแนวคิดในการแบ่งช่วงค่าที่แตกต่างกันไป ในที่นี้จะยกตัวอย่างเพียง 3 คลาส คือ Latin Hypercubes Design (LHD) Hamersley Sequence Sampling (HSS) Uniform Design

#### Latin Hypercubes Design (LHD)

ใช้ครั้งแรกโดย McKey et al. ใน 1979 เป็นเมทริกซ์ (X) มีมิติ  $n \times d$  โดยที่  $n$  คือจำนวนรัน และ  $d$  คือจำนวนตัวแปรเข้า ซึ่งแต่ละหลัก  $d$  ประกอบด้วยรันตั้งแต่  $\{1, 2, \dots, n\}$  ที่นำมาเรียงสับเปลี่ยนกัน โดยค่าเหล่านี้จะเกิดจากการเปลี่ยนจากค่าที่แท้จริงให้อยู่ในรูปแบบตัวแทนตามที่กำหนด ในการทำงานผู้ออกแบบสามารถเทียบค่าตัวแทนกับค่าจริงของแต่ละตัวแปรได้โดยง่ายผ่านสูตร การแบ่งช่วงของ design space แบบนี้จะได้ค่าที่ไม่ซ้ำกัน ตามรูปแบบของ Latin Hypercubes ค่าในแต่ละช่วงหาได้จากสูตรหลัก แต่ที่นิยมนำมาประยุกต์ใช้นั้นจะเป็นแบบที่แบ่งค่าจริงให้อยู่ในช่วง  $[0,1]$  เพื่อความง่าย สะดวกและรวดเร็วในการคำนวณ โดยผู้ออกแบบสามารถเทียบค่าตัวแทนไปเป็นค่าจริงได้ง่าย

LHD สามารถสร้างได้จากแนวคิดของ Stratified sampling (McKey, Beckman and Conover, 1979) โดยแนวคิดนี้จะทำให้มั่นใจได้ว่าโดยทุกๆ ส่วนย่อยของ design space จะมีการแบ่งตัวแปรเข้าในระดับที่เท่าๆ กันทุกตัว และมีความเป็นไปได้เท่าๆ กัน โดยโอกาสที่แต่ละตัวแทนจะถูกเลือกบน Latin hypercubes เป็นดังสมการต่อไปนี้

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

โดยที่  $\pi_{ij}$  เป็นส่วนของเมทริกซ์  $n \times d$  ที่ประกอบด้วยสตอม์  $\pi_i (i=1,2,\dots,d)$  โดยแต่ละสตอม์  $\pi_i (i=1,2,\dots,d)$  จะสุ่มเรียงสลับสับเปลี่ยนโดยอิสระด้วย  $1,2,\dots,n$  และ  $U_{ij}$  คือค่า  $n \times d$  สำหรับ i.i.d.uniform  $U[0,1]$  สุ่มค่าโดยอิสระของ  $\pi_{ij}$  ดังแสดงในตารางที่ 1 ซึ่งจะเห็นว่าค่าของตัวแปรทั้ง 4 ตัว จะมีการแบ่งเป็นช่วงที่เท่ากัน และโดยในแต่ละสตอม์จะเป็นการเรียงสลับเปลี่ยนของค่าตัวแปรแต่ละตัว

ตาราง 1 ตัวอย่าง LHD 4x5 แบบสุ่ม

| รันที่ | X1 | X2 | X3 | X4 |
|--------|----|----|----|----|
| 1      | 1  | 2  | 4  | 3  |
| 2      | 2  | 4  | 1  | 4  |
| 3      | 3  | 5  | 3  | 1  |
| 4      | 4  | 3  | 2  | 2  |
| 5      | 5  | 1  | 5  | 5  |

#### Hammersley Sequence Sampling (HSS)

Hammersley Sequence Sampling เป็นการออกแบบการทดลองที่เหมาะสมกับแผนการทดลองที่มีคลาดเคลื่อนต่ำ โดยคลาดเคลื่อนต่ำในที่นี้หมายถึงความเป็นอันหนึ่งอันเดียวกันของการกระจายของจุดใน design space โดย Hammersley Sequence Sampling จะมีการกำหนดจุด  $n$  ในมิติ  $k$  ของ hypercube (Kilgus and Diwekar, 1997) โดยจะเน้นในเรื่องของการเตรียมคุณสมบัติความเป็นอันหนึ่งอันเดียวกันบน  $k$  มิติ มากกว่า Latin hypercubes

#### Uniform Design

Uniform Design เป็นการออกแบบที่เตรียมจุดออกแบบให้มีการกระจายอย่างเดียวกันในการออกแบบการทดลองแบบ Uniform Design นั้นจัดเป็นรูปแบบหนึ่งของการออกแบบแบบ fractional factorial design โดยในที่นี้จะเพิ่มคุณสมบัติความเป็นแบบเดียวกันเข้าไป Uniform Design ถูกใช้อย่างแพร่หลายตั้งแต่ปี 1980 (Fang, 1980)

แผนการทดลองแบบ Uniform Design และ Latin hypercubes จะคล้ายคลึงกัน แต่ถ้าพิจารณาภาพรวมในทุกๆ มิติของตัวแปรเข้า แผนการทดลองแบบ Uniform Design จะมี

คุณสมบัติการกระจายจุดทดลองที่ดีกว่า Latin hypercubes การออกแบบทั้งสองแบบนี้จะมีความแตกต่างกันมากขึ้นเมื่อจำนวนมิติมากขึ้น

### กล่องดำ หรือ Black box

Simulation Routine หรือ กล่องดำ (black box) คือ กระบวนการทำงานที่เราไม่สามารถมองเห็นและคาดเดาได้ว่าเกิดอะไรขึ้นภายในหรือกลไกภายในเป็นอย่างไร กระบวนการใน black box นี้จะใช้รูปแบบสมการทางคณิตศาสตร์ที่มีความซับซ้อนหรือเป็นการจำลองการทำงานโดยใช้กลไกแบบ Finite State Machine หรือแบบอื่นๆ เพื่อคำนวณผลลัพธ์ออกมา ซึ่งผลลัพธ์ที่ได้นี้จะแปรผันตามระดับค่าต่างๆ ของปัจจัยที่เกี่ยวข้องที่ป้อนเข้าไป โดยส่วนมากกล่องดำจะถูกศึกษาและพัฒนาขึ้นโดยผู้เชี่ยวชาญทางด้านนั้นๆ เช่น กล่องดำจำลองการทำงานของปฏิกิริยาปฏิกิริยาจำลองการทำงานของวงจรอิเล็กทรอนิกส์ เป็นต้น ซึ่งจะเห็นว่าระบบเหล่านี้จะมีความซับซ้อนภายในอย่างมาก เพราะเป็นการจำลองปรากฏการณ์ที่ยากต่อการคาดเดา

### Approximation Models

เมื่อได้การออกแบบการทดลองมา 1 ชุด ก็จะนำชุดการทดลองนี้ไปเป็นค่าที่ส่งเข้าไปในกล่องดำ และได้ค่าผลลัพธ์ออกมาตามชุดทดลอง ค่าผลลัพธ์เหล่านี้จะเป็นตัวแทนของโมเดลที่ผู้ศึกษาสนใจ โมเดลที่จะสร้างนั้นเกิดจากการนำค่าตัวแปรเข้าจากชุดทดลอง และค่าผลลัพธ์ตอบสนองเหล่านี้มาใส่เข้ากับหลักการการสร้างโมเดล ซึ่งมีด้วยการหลายหลักการ โดยในที่นี้จะนำเสนอเพียง 2 หลักการที่เป็นที่นิยมใช้กัน

#### Response surfaces

การพัฒนาโมเดลตามแบบ Response surfaces นั้นพัฒนาขึ้นในครั้งแรกเพื่อวิเคราะห์การทดลองทางกายภาพ (Box and Wilson, 1951) เป็นหลัก ดังเช่น Polynomial response surface models ถูกใช้อย่างแพร่หลายและอย่างมีประสิทธิภาพ เพื่อสร้างโมเดลประมาณในงานประยุกต์ โดยจะมีรูปแบบสมการโมเดลดังตัวอย่างต่อไปนี้ เช่น สมการ second-order polynomial response surface models จะมีรูปแบบสมการ คือ

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

ซึ่งพารามิเตอร์  $\beta$  คำนวณได้จาก Least squares regression ส่วน  $\hat{y}$  เป็นค่าทำนายจากการเบี่ยงเบนของกำลังสองของผลบวกที่น้อยที่สุดของ Least squares regression จากค่าจริง  $y(x)$  โดยใช้สมการ

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

โดย  $X$  คือ เมตริกซ์การวางแผนสำหรับจุดตัวอย่าง  $X'$  คือ ทราบดีของ  $X$  และ  $y$  คือเวกเตอร์หลักที่ประกอบด้วยค่าตอบสนองของแต่ละจุดตัวอย่าง Polynomial response surface models สร้างได้ง่ายและมีความสามารถคำตอบที่เหมาะสมได้ราบเรียบเร็วแม้จะมีสัญญาณรบกวน

### Kriging

การออกแบบโมเดลแบบ Kriging นั้นพัฒนาขึ้นเพื่อประยุกต์ใช้ในงานด้านธรณีสถิติ (geostatistics) โดย Kriging โมเดลจะสมมติการจัดกลุ่มสำหรับ polynomial model และแยกแต่ละกลุ่มไปตามสมการดังต่อไปนี้

$$\hat{y} = \sum_{j=1}^k \beta_j f_j(x) + Z(x)$$

โดย  $Z(x)$  ถูกสมมติให้เป็นค่าจริงของกระบวนการสุ่มด้วยค่าศูนย์กลางและฟังก์ชันความสัมพันธ์เชิงระยะทาง (spatial correlation function) ตามสมการต่อไปนี้

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

โดย  $\sigma^2$  คือ ความแตกต่างของกระบวนการ ฟังก์ชันความสัมพันธ์โดยเราสามารถเลือกใช้ได้หลากหลายวิธี แต่วิธีที่นิยมและถูกเลือกใช้โดยทั่วไปคือวิธี Gaussian correlation (Sack, 1989)

### อัลกอริทึมการค้นหา

การออกแบบการทดลองที่ดีจะส่งผลให้ได้โมเดลจำลองที่มีความแม่นยำและมีประสิทธิภาพในการพยากรณ์ การออกแบบการทดลองนั้นจะถูกจำกัดด้วยจำนวนตัวแปรและจำนวนครั้งในการรัน เพราะการทำการทดลองในแต่ละรันนั้นใช้เวลาและเงินจำนวนมาก การออกแบบการทดลองที่ดีที่สุดจะมาจาก "การเลือกทางเลือกที่ดีที่สุด" ซึ่งจะเรียกว่าการทำงานแบบ Optimization

การจำลองการทดลองด้วยคอมพิวเตอร์นั้นเป็นการทำงานแบบแน่นอน Deterministic ดังนั้นการออกแบบจึงเน้นไปที่การเลือกแผนการทดลอง (Experimental design) ที่สามารถครอบคลุม design space ให้ได้ทั่วถึงมากที่สุด โดยจากที่กล่าวมาจะเห็นว่าทางเลือกที่เป็นไปได้ทั้งหมดนั้นมีจำนวนมากมายมหาศาล การนำอัลกอริทึมการค้นหา (Search algorithm) มาประยุกต์ใช้จะเกิดประโยชน์อย่างมากในการช่วยค้นหาทางเลือกที่ดีที่สุด

ปัญหาดังกล่าวนี้อยู่ในกลุ่มของปัญหาที่เรียกว่า Optimization problem ซึ่งเป็นลักษณะการหาคำตอบที่เหมาะสมที่สุดหรือที่ยอมรับได้จากคำตอบที่เป็นไปได้ทั้งหมดที่มีจำนวนมากมายมหาศาล ตัวอย่างปัญหาในกลุ่มนี้ได้แก่ Traveling salesman problem: TSP, Time schedule เป็นต้น

อัลกอริทึมการสืบค้นที่สามารถนำมาใช้ค้นหาแผนการทดลองนั้นมีด้วยกันหลายวิธี ได้แก่ Simulated annealing: SA, Genetic algorithms: GA, Tabu search: TA เป็นต้น โดยหลักการสืบค้นจะทำงานควบคู่ไปกับเกณฑ์ที่ใช้ในการวางแผนการทดลองที่เหมาะสม (Optimality criteria) ในที่นี้จะกล่าวถึง 2 อัลกอริทึมการสืบค้น คือ GA และ SA

### อัลกอริทึมการสืบค้นแบบเจเนติก – Genetic Algorithm (GA)

อัลกอริทึมการสืบค้นแบบเจเนติก เป็นการทำงานที่จำลองขั้นตอนวิวัฒนาการทางพันธุกรรมของสิ่งมีชีวิต ขั้นตอนวิธีเชิงพันธุกรรม (genetic algorithm) เป็นเทคนิคสำหรับค้นหาผลเฉลย (solutions) หรือคำตอบโดยประมาณของปัญหา โดยอาศัยหลักการจากทฤษฎีวิวัฒนาการจากชีววิทยา และ การคัดเลือกตามธรรมชาติ (natural selection) นั่นคือ สิ่งมีชีวิตที่เหมาะสมที่สุดก็จะอยู่รอด กระบวนการคัดเลือกได้เปลี่ยนแปลงสิ่งมีชีวิตให้เหมาะสมยิ่งขึ้น ด้วยตัวปฏิบัติการทางพันธุกรรม (genetic operator) เช่น การสืบพันธุ์ (inheritance หรือ reproduction) การกลายพันธุ์ (mutation) การแลกเปลี่ยนยีน (recombination) เป็นต้น

ขั้นตอนวิธีเชิงพันธุกรรมเป็นการจำลองทางคอมพิวเตอร์ เพื่อแก้ปัญหาค่าที่เหมาะสมที่สุด (optimal solution) โดยการแทนคำตอบที่มีอยู่ให้อยู่ในลักษณะ โครโมโซม (chromosomes) แล้วปรับปรุงคำตอบแต่ละชุด (เรียกว่า individual) ด้วยวิธีการต่างๆ ซึ่งเกี่ยวข้องกับการวิวัฒนาการ (evolutionary operation) อาจเป็นการเปลี่ยนแปลงยีนแบบสุ่ม ด้วยตัวปฏิบัติการทางพันธุกรรม (evolutionary operator) เพื่อให้ได้คำตอบที่ดีขึ้น โดยทั่วไปจะแทนคำตอบด้วยเลขฐานสอง (สายอักขระของเลข 0 และ 1)

การวิวัฒนาการ (evolution) เพื่อหาคำตอบที่เหมาะสมที่สุด (the fitness solution) จะเริ่มจากการสร้างประชากรที่ได้จากการสุ่มทั้งหมดและจะทำการเป็นรุ่นๆ โดยในแต่ละรุ่นคำตอบ คำตอบหลายชุดจะถูกสุ่มเลือกขึ้นมาเปลี่ยนแปลง ซึ่งอาจจะทำให้เกิดการกลายพันธุ์ หรือสืบเปลี่ยนยีนระหว่างกัน จนได้ประชากรรุ่นใหม่ที่มีค่าความเหมาะสม (fitness) มากขึ้น การวิวัฒนาการนี้จะทำไปเรื่อยๆ จนกระทั่งพบคำตอบที่มีค่าความเหมาะสมตามต้องการ (wikipedia, 2007)

GA ที่จะแสดงเป็นการประยุกต์ GA นำเสนอโดย Liefvendel และ Stocki (2006) เพื่อสืบค้นหาแผนการทดลองที่เหมาะสมที่สุดบน LHD จะปรับปรุงวิธีการสร้างทายาท (generating offsprings) และวิธีการกลายพันธุ์ (generating mutation) GA จะเริ่มต้นด้วยการสร้างประชากรเริ่มต้นประกอบด้วย mLHD สำหรับเมทริกซ์  $nx d$  รุ่นใหม่จะถูกสร้างโดยการ crossover และประยุกต์การกลายพันธุ์ การค้นสืบค้นจะทำงานไปเรื่อยๆ จะหยุดเมื่อตรงกับเกณฑ์ ขั้นตอนโดย

ละเอียดมีดังนี้ จะให้  $m$  คือ ขนาดของประชากร,  $m_i$  คือ จำนวนของรูปแบบที่สร้างในแต่ละรุ่น ( $m_i = 10 \times m$ ),  $\omega$  คือ อัตราการกลายพันธุ์

ขั้นที่ 1: กำหนดประชากรเริ่มต้น สร้าง  $m$  จากหลักการสุ่มเลือก LHD ขนาด  $n \times d$  โดยกำหนดจำนวนประชากรเริ่มต้นเป็น  $m$  ดังนั้นประชากรเริ่มต้นจะประกอบด้วย  $\{X^i; i = 1, 2, \dots, m\}$  LHD

ขั้นที่ 2: ให้  $P$  เป็น  $\{X^i; i = 1, 2, \dots, m\}$  LHD ที่เรียงลำดับตามเกณฑ์ค่าความเหมาะสม

ขั้นที่ 3: เลือกรุ่นพ่อแม่ โดยการสุ่มเลือก LHD มา 2 ตัวจากขั้นที่ 1 โดยเรียกว่า  $X^a$  และ  $X^b$  จากนั้นให้เลือกหนึ่งหลักของแต่ละ LHD เรียกว่า  $C^a$  และ  $C^b$  ดังรูปที่ 6

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 2 | 7 |
| 2 | 1 | 4 | 5 |
| 3 | 4 | 6 | 3 |
| 4 | 3 | 1 | 1 |
| 5 | 6 | 3 | 6 |
| 6 | 5 | 5 | 4 |
| 7 | 7 | 7 | 2 |

$C_a$

$X^b$

|   |   |   |   |
|---|---|---|---|
| 3 | 3 | 4 | 7 |
| 4 | 2 | 3 | 3 |
| 6 | 1 | 6 | 2 |
| 5 | 4 | 5 | 1 |
| 7 | 6 | 7 | 4 |
| 1 | 7 | 1 | 6 |
| 2 | 5 | 2 | 5 |

$C_b$

รูปที่ 6 แสดงการเลือกหลัก  $C^a$  และ  $C^b$  จาก  $X^a$  และ  $X^b$

ขั้นที่ 4: สร้างหลักของทายาท จากหลัก  $C^a$  และ  $C^b$  จะได้หลักรุ่นทายาทเป็น  $O^a$  และ  $O^b$  ตามหลักการสร้างต่อไปนี้

- ระบุแถวที่มีค่าเหมือนกันในแต่ละหลัก  $C^a$  และ  $C^b$  แล้วทำการคัดลอกค่าดังกล่าวไปไว้ที่  $O^a$  และ  $O^b$  ดังรูปที่ 7 ที่ทำเป็นเส้นประสีเหลี่ยม
- สุ่มค่า 2 ค่าเพื่อคัดลอกตำแหน่งแถวในหลัก  $C^a$  และ  $C^b$  มาสองตำแหน่ง  $q$  และ  $r$  จากนั้นให้คัดลอกค่าตั้งแต่ตำแหน่ง  $q$  ถึง  $r$  ไปยัง  $O^a$  และ  $O^b$  ดังรูปที่ 7
- ใส่ค่าลงตำแหน่งที่เหลือของ  $O^a$  ด้วยค่าที่เรียงตามที่ปรากฏบน  $C^b$  และให้ทำงานลักษณะเดียวกันกับ  $O^b$  คือใส่ค่าที่เหลือของ  $O^b$  ด้วยค่าที่เรียงตามที่ปรากฏบน  $C^a$  ดังรูปที่ 7

|     | $Ca$ | $Cb$ | $I$ | $C$ | $O^b$ |
|-----|------|------|-----|-----|-------|
|     | 2    | 3    |     |     | 2     |
|     | 4    | 4    | 4   | 4   | 4     |
| $q$ | 6    | 6    | 6   | 6   | 6     |
|     | 1    | 5    |     | 5   | 5     |
|     | 3    | 7    |     | 7   | 7     |
| $r$ | 5    | 1    |     | 1   | 1     |
|     | 7    | 2    |     |     | 3     |

|     | $Ca$ | $Cb$ | $I$ | $C$ | $O^a$ |
|-----|------|------|-----|-----|-------|
|     | 2    | 3    |     |     | 7     |
|     | 4    | 4    | 4   | 4   | 4     |
| $q$ | 6    | 6    | 6   | 6   | 6     |
|     | 1    | 5    |     | 1   | 1     |
|     | 3    | 7    |     | 3   | 3     |
| $r$ | 5    | 1    |     | 5   | 5     |
|     | 7    | 2    |     |     | 2     |

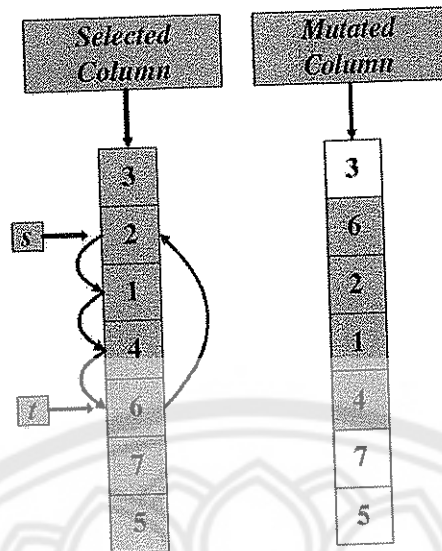
รูปที่ 7 แสดงการสร้างหลักรุ่นทายาท  $O^a$  และ  $O^b$

ขั้นตอน 5: สร้างสายพันธุ์ใหม่หรือทายาท (Offspring) ให้สุ่มเลือกหลักจาก  $X^a$  และ  $X^b$  แล้วนำหลักรุ่นทายาท  $O^a$  ไปแทนที่ ทำซ้ำอีกครั้งแต่ให้นำหลักรุ่นทายาท  $O^b$  มาแทนที่ จะทำให้ได้ LHD ใหม่มาทั้งหมด 4 ตัว

ขั้นตอน 6: ทำซ้ำขั้นตอน 3, 4 และ 5 จนกระทั่งมีจำนวนทายาทอย่างน้อยหรือเท่ากับ  $m_i$

ขั้นตอน 7: ปฏิบัติการการกลายพันธุ์บนทายาทที่ได้มา  $m_i$  ตัว ที่ได้จากขั้นตอน 6

- สุ่มเลือกหลักเพื่อกลายพันธุ์โดยให้ความน่าจะเป็นเท่ากับ  $\omega$
- ทำการกลายพันธุ์บนหลักที่ได้สุ่มเลือกไว้แล้ว โดยการสุ่มตัวเลขมา 2 ตัว คือ  $s$  และ  $t$  เพื่อเป็นการระบุตำแหน่งแถวที่จะมีการเลื่อนหรือสลับค่าแบบการเรียงแบบหมุน (cyclic permutation) ดังรูปที่ 8



รูปที่ 8 แสดงการกลายพันธุ์ในแต่ละทายาทที่ถูกคัดเลือก

ขั้นตอน 8: เรียงลำดับทายาท LHD ทั้ง  $m$  ตัวตามเกณฑ์หาค่าเหมาะสม (Optimality criteria) ที่ได้จากขั้นตอน 7 จากนั้นทำการจัดเก็บ LHD ที่ดีที่สุดไว้จำนวน  $(m-2)$  ตัว กล่าวคือ จะได้  $\{X_i^* : i=1,2,\dots,m-2\}$  แล้วเลือก LHD ที่ดีที่สุด 2 ตัวมาจากขั้นตอน 2 มารวมโดยตั้งให้เป็นตัวที่  $X_{m-1}^*, X_m^*$  ดังนั้นจะได้ทายาท LHD ในรุ่นนี้ทั้งหมด  $m$  ตัว กล่าวคือ  $\{X_i^* : i=1,2,\dots,m\}$

ขั้นตอน 9: ให้  $\{X_i : i=1,2,\dots,m\} = \{X_i^* : i=1,2,\dots,m\}$  ทำซ้ำขั้นตอน 2 ถึง ขั้นตอน 8 ไปจนกว่าจะหยุดด้วยค่าการหยุดที่กำหนดไว้ดังนี้

1. หาค่าเริ่มต้นของค่าเกณฑ์หาค่าเหมาะสมในที่นี้คือ  $\phi_p$  (Phi P)

$$\Delta(I) = \phi_p(0) - \phi_p(I)$$

โดย  $\phi_p(0)$  และ  $\phi_p(I)$  เป็นค่า  $\phi_p$  ของการออกแบบที่ดีที่สุดในรอบแรกของรุ่นประชากร และในรอบที่  $I$  ของรุ่นการทดลอง

2. GA จะหยุดการทำงานเมื่อ

$$\phi_p(I) - \phi_p(k) < \epsilon \Delta(I)$$

โดย  $\epsilon$  คือค่าที่กำหนดขึ้นเพื่อใช้ควบคุมเงื่อนไขการหยุด

เมื่อหยุดจากการวนซ้ำให้รายงาน LHD ที่ดีที่สุดตามเกณฑ์หาค่าเหมาะสม (Optimality criteria) ที่ได้เก็บไว้

## อัลกอริทึมการสืบค้นแบบซิมูเลทเทค แอนนิลลิ่ง – Simulated Annealing (SA)

Morris และ Mitchell (Morris and Mitchell, 1995) ได้ดัดแปลงอัลกอริทึมสืบค้น Simulated Annealing เพื่อหาแผนการทดลองที่ดีที่สุด โดยใช้เกณฑ์ค่าที่เหมาะสมที่สุดควบคู่ไปด้วยคือ เกณฑ์  $\phi_p$  แนวคิดของอัลกอริทึม SA เป็นหลักการสลับค่าสองจุดในแผนการทดลองโดยการสุ่ม (Perturbation) ของมิติในแผนการทดลอง อย่างไรก็ตามในอัลกอริทึม SA ที่มีการสลับค่าจะบอกไม่ได้ว่าดีขึ้นหรือแย่ลง ต้องใช้เกณฑ์  $\phi_p$  ในขั้นตอนสืบค้นเป็นตัวบอก ขั้นตอนของอัลกอริทึม SA มีดังนี้

ขั้นตอน 1: กำหนดค่าเริ่ม  $I_{max}$  (จำนวนมากที่สุดของการสลับแล้วปรับปรุงดีขึ้น),  $t_0$  (อุณหภูมิเย็นเริ่มต้น),  $C_t$  (ตัวประกอบให้ลด  $t_0$  เมื่อสลับแล้วไม่ได้ดีขึ้นในหลักเกณฑ์  $\phi_p$ ) ค่า  $t_0$  จะเป็นตัวควบคุมความน่าจะเป็นในการสลับแล้วไม่ได้ปรับปรุงดีขึ้น กระบวนการสืบค้นดำเนินการไปเรื่อย ถ้าไม่มีการปรับปรุงให้ดีขึ้นก็จะลดอุณหภูมิ โดยคูณ  $t_0$  ด้วย  $C_t$

ขั้นตอน 2: สุ่มสร้าง LHD ขนาด  $n \times d$  แล้วให้  $X_{best} = X, t = t_0$

ขั้นตอน 3: กำหนด  $I = 1, Counter = 0$

ขั้นตอน 4: กำหนด  $X_{try} = X$  แล้วสุ่มเลือกหลัก ( $j$ ) ของเมทริกซ์  $X_{try}$  และสุ่มสองค่าในหลัก ( $j$ ) แล้วสลับค่ากัน เรียกว่า  $X_{aj} \leftrightarrow X_{bj}$

ขั้นตอน 5: กำหนด  $X = X_{try}, Counter = 1$

$$\text{ถ้า } \phi_p(X) - \phi_p(X_{try}) \geq \varepsilon$$

$$\text{หรือความน่าจะเป็นเป็น } e^{-(\phi_p(X_{try}) - \phi_p(X))/t} \text{ เมื่อ } \phi_p(X) - \phi_p(X_{try}) < \varepsilon$$

ขั้นตอน 6: ถ้า  $\phi_p(X_{try}) < \phi_p(X_{best})$  แล้วกำหนด  $I = 1$  และ  $X_{best} = X_{try}$

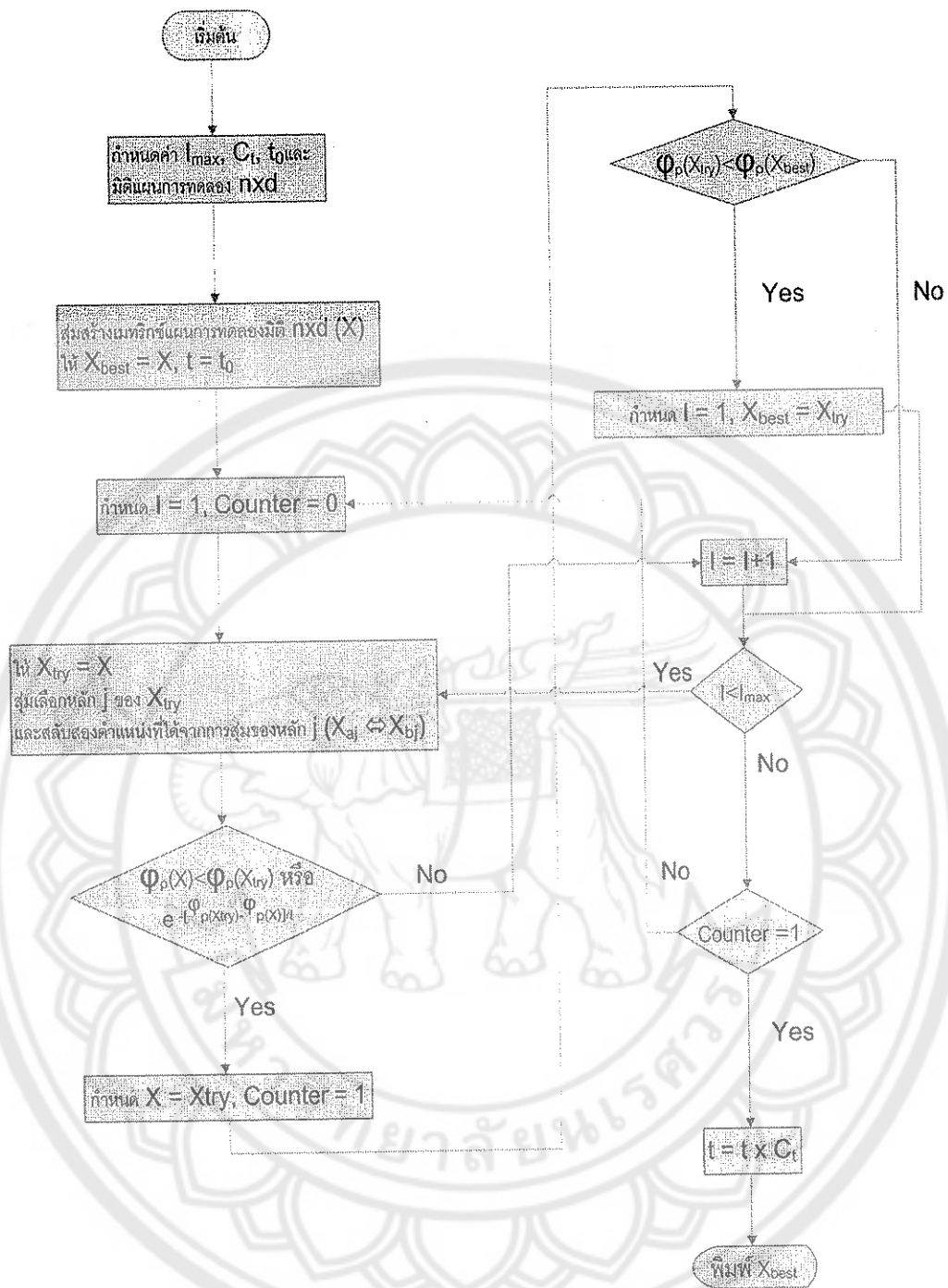
$$\text{ถ้าไม่ให้ } I = I + 1$$

ขั้นตอน 7: ถ้า  $I < I_{max}$  ให้กลับไปขั้นตอน 4

ขั้นตอน 8: ถ้า  $Counter = 1$  ให้กำหนด  $t = t \times C_t$  แล้วกลับไปขั้นตอน 3

ขั้นตอน 9: จบกระบวนการและรายงานค่า  $X_{best}$

ขั้นตอนทั้ง 9 นั้นสามารถสรุปเป็นผังงานได้ดังรูปที่ 9



รูปที่ 9 ผังงานอัลกอริทึมการสืบค้นแบบซิมูเลตเตด แอนนิลลิง – Simulated Annealing (SA)

### บทที่ 3

#### วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงวิธีการดำเนินการวิจัยซึ่งมีรายการดังนี้

- 1) กำหนดเกณฑ์หาค่าเหมาะสม (Optimality criteria)
- 2) กำหนดค่าพารามิเตอร์ต่างๆ สำหรับอัลกอริทึมการค้นหาทั้ง 2 วิธี
- 3) ออกแบบการทดลอง
- 4) สรุปผลการทดลอง

#### กำหนดเกณฑ์หาค่าเหมาะสม (Optimality criteria)

การกำหนดเกณฑ์หาค่าเหมาะสมนั้นเป็นปัจจัยหลักของการทำงานของอัลกอริทึมการค้นหา สำหรับงานวิจัยนี้จะเลือกใช้เกณฑ์หาค่าเหมาะสมที่ชื่อว่า  $\phi_p$  (Phi P) ซึ่งคิดค้นโดย Morris และ Mitchell (Morris and Mitchell, 1995) เพื่อใช้เป็นเกณฑ์หาค่าเหมาะสมสำหรับการออกแบบแผนการทดลอง โดยหลักการหาค่า  $\phi_p$  นั้นจะแนวคิดดังต่อไปนี้

- กำหนดให้  $X$  เป็นเมตริกซ์ใดๆ ที่ถูกกำหนดขึ้นโดยกำหนดขนาดเป็น  $n \times d$  โดย  $n$  คือจำนวนรันในแผนการทดลอง และ  $d$  คือจำนวนตัวแปรซ้ำของแผนการทดลองจะได้

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}$$

- จากนั้นให้คำนวณหาค่าระยะห่างยูคลิดีน (Euclidean distance) ระหว่างจุด  $X_i$  และ  $X_j$  ใดๆ จากสมการนี้

$$d(X_i, X_j) = \left[ \sum_{k=1}^d (X_{ik} - X_{jk})^2 \right]^{\frac{1}{2}}$$

- จะได้ชุดข้อมูลที่เก็บค่าระยะห่าง  $(d_1, d_2, \dots, d_m)$  โดยเป็นค่าระยะห่างที่เรียงลำดับจากค่าน้อยไปมาก และให้  $(J_1, J_2, \dots, J_m)$  โดย  $J_j$  เป็นจำนวนคู่ในจุดทดลองที่มีระยะห่าง เป็น  $d_j$

- คำนวณค่า  $\phi_p$  ได้จากสมการต่อไปนี้

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

- ค้นหาที่น้อยที่สุดในชุดนี้ค่าที่ได้จะเป็นค่า  $\phi_p$  ต่ำที่สุดซึ่งเป็นค่าที่เราต้องการ

กำหนดค่าพารามิเตอร์ต่างๆ สำหรับอัลกอริทึมการค้นหาทั้ง 2 วิธี

การตั้งค่าพารามิเตอร์ของอัลกอริทึมการสืบค้นแบบเจนเนติก

- $m = 50$
- $m_i = 10 \times m = 500$
- $\omega = 0.05$
- $\epsilon = 0.0001$

การตั้งค่าพารามิเตอร์ของอัลกอริทึมการสืบค้นแบบชิมูเลตเตด แอนนิลลิง

- กำหนดให้ค่าเริ่มต้นของ  $t_0$  เกิดจากค่าเฉลี่ยของระยะทางยูคลิดีน (Euclidean distance) ระหว่างจุดใน LHD เริ่มต้น  $X$

$$t_0 = \sum_{i=1}^n \sum_{j=i+1}^n \left[ \sum_{l=1}^d (X_{il} - X_{jl})^2 \right]^{1/2} / \binom{d}{2}$$

- $I_{\max}$  คือจำนวนการสับเปลี่ยนที่มากที่สุดก่อนที่จะมีการลดอุณหภูมิ โดยค่านี้จะต้องมากพอที่จะทำให้กลไกของ SA หลุดออกจากค่าที่ดีที่สุดในระดับกลุ่ม (local minimum) ในที่นี้  $I_{\max} = 1000$  สำหรับ  $d \leq 6$  และ  $I_{\max} = 500$  สำหรับ  $d \geq 7$
- $C_r$  คืออัตราในการลดอุณหภูมิ โดยกำหนดให้เท่ากับ 0.95 เพื่อค่อยๆ ลดอุณหภูมิและทำให้การค้นหาเป็นไปอย่างช้าๆ
- $\epsilon$  คือค่าความต่าง (tolerance) ที่ตั้งไว้ที่  $1.0E-15$  สำหรับ  $d \leq 6$  และเท่ากับ 0.0001 สำหรับ  $d \geq 7$



### ออกแบบการทดลอง

ในการทดสอบเพื่อเปรียบเทียบประสิทธิภาพการอัลกอริทึมการสืบค้นของทั้ง 2 วิธีนี้ผู้วิจัย ดำเนินการทดสอบ  
ได้ทดสอบการทำงานบนมิติปัญหาที่กำหนดไว้ดังตารางต่อไปนี้

- 5 JUL 2011

| $d$ | 2 | 3  | 4  | 5  | 6  | 7  | 8   | 9   | 10  |
|-----|---|----|----|----|----|----|-----|-----|-----|
| $n$ | 9 | 19 | 33 | 51 | 73 | 99 | 129 | 163 | 201 |

พัฒนาโปรแกรมด้วยโปรแกรม R และทำการทดลองการทำงานในแต่ละกรณีทั้งหมด 30 ครั้ง  
บันทึกผลลัพธ์เพื่อนำไปวิเคราะห์ต่อไป

### สรุปผลการทดลอง

นำค่าที่ได้จากการทดลองมาวิเคราะห์หาค่าเฉลี่ย ค่าส่วนเบี่ยงเบนและทดสอบเปรียบเทียบ  
ด้วยตัวสถิติทดสอบ t-test ได้บทสรุปดังตารางที่ 2

ตารางที่ 2 แสดงผลการทดสอบประสิทธิภาพการทำงานระหว่าง SA และ GA

| $d$ | $n$ | Algorithm | Min   | Max   | Mean  | SD    | t-test  | P-value |
|-----|-----|-----------|-------|-------|-------|-------|---------|---------|
| 2   | 9   | SA        | 4.274 | 4.350 | 4.324 | 0.033 | -3.716  | 0.002   |
|     |     | GA        | 4.345 | 4.509 | 4.405 | 0.057 |         |         |
| 3   | 19  | SA        | 4.914 | 4.949 | 4.931 | 0.013 | -16.497 | 0.000   |
|     |     | GA        | 5.344 | 5.696 | 5.485 | 0.100 |         |         |
| 4   | 33  | SA        | 5.197 | 5.205 | 5.202 | 0.002 | -37.392 | 0.000   |
|     |     | GA        | 5.670 | 5.820 | 5.760 | 0.045 |         |         |
| 5   | 51  | SA        | 5.415 | 5.423 | 5.419 | 0.003 | -48.790 | 0.000   |
|     |     | GA        | 6.051 | 6.182 | 6.115 | 0.043 |         |         |
| 6   | 73  | SA        | 5.582 | 5.585 | 5.583 | 0.001 | -32.593 | 0.000   |
|     |     | GA        | 6.236 | 6.454 | 6.334 | 0.069 |         |         |
| 7   | 99  | SA        | 5.733 | 5.735 | 5.734 | 0.001 | -52.450 | 0.000   |
|     |     | GA        | 6.246 | 6.341 | 6.289 | 0.032 |         |         |
| 8   | 129 | SA        | 6.166 | 6.197 | 6.185 | 0.011 | -32.191 | 0.000   |
|     |     | GA        | 6.354 | 6.391 | 6.371 | 0.014 |         |         |
| 9   | 163 | SA        | 6.336 | 6.404 | 6.374 | 0.019 | -9.330  | 0.000   |
|     |     | GA        | 6.423 | 6.449 | 6.438 | 0.007 |         |         |
| 10  | 201 | SA        | 6.480 | 6.523 | 6.501 | 0.013 | -11.664 | 0.000   |
|     |     | GA        | 6.565 | 6.642 | 6.609 | 0.024 |         |         |

จากตารางที่ 2 ค่าที่วัดเพื่อนำมาประเมินคือค่า  $\phi_p$  โดยค่า  $\phi_p$  ยิ่งน้อยจะแสดงถึงแผนการทดลองที่ดี  
ดังนั้นค่าน้อยยิ่งดีสำหรับการเปรียบเทียบในงานวิจัยนี้ จากตารางจะเห็นว่า SA จะได้ค่าเฉลี่ย  
ค่าสูงสุด และค่าต่ำสุดที่น้อยกว่า GA ในทุกการทดลองที่  $d$  และ  $n$  แสดงว่า SA สามารถหา

แผนการทดลองที่ดีกว่าที่ GA คำหามา นอกจากนี้จากค่าสถิติทดสอบ t-test และ p-value ซึ่งมีค่าน้อยกว่า 0.05 เป็นการยืนยันว่ามีความแตกต่างอย่างมีนัยสำคัญที่ความเชื่อมั่นร้อยละ 95 จากตารางที่ 2 นอกจากแสดงให้เห็นว่า SA หาคำตอบที่ดีกว่า GA แล้วค่า SD ที่ต่ำของ SA ยังบ่งบอกถึงลักษณะการลู่เข้าหาคำที่ดีที่สุดของ SA นั้นดีกว่า GA



## บทที่ 4

### สรุปผลการดำเนินงาน

#### สรุปผลการดำเนินงาน

สิ่งที่ได้จากงานวิจัยนี้คือการประยุกต์ใช้ SA กับการออกแบบการทดลองด้วยคอมพิวเตอร์นั้นจะให้คำตอบที่ดีกว่า GA ด้วยเทคนิคการสร้างสายพันธุ์และการกลายพันธุ์ดังที่ระบุในงานวิจัยนี้ ภายใต้เกณฑ์วัดค่าที่ดีที่สุด  $\phi_p$

นอกจากนี้ยังสรุปได้ว่า SA ใช้เวลาในการค้นหาน้อยกว่า GA และยังมีความซับซ้อนในกลไกน้อยกว่า อย่างไรก็ตามภายใต้เทคนิคการสร้างสายพันธุ์และการกลายพันธุ์ที่แตกต่างออกไป อาจส่งผลให้ GA ค้นหาคำตอบได้ดียิ่งขึ้น

ในงานวิจัยนี้ไม่ได้ทำการทดลองในมิติที่ขนาดใหญ่เกิน  $10 \times 201$  ซึ่งอาจจะได้ผลลัพธ์ที่แตกต่างออกไปแต่ต้องใช้เวลาในการทดลองนานยิ่งขึ้น

เนื่องจาก SA ได้แสดงให้เห็นถึงประสิทธิภาพที่โดดเด่นในการค้นหาการทดลองที่ดีกว่า GA ในงานวิจัยต่อไปผู้วิจัยจึงเล็งเห็นความเป็นไปได้ที่จะพัฒนาให้ SA ทำงานได้ดีและเร็วยิ่งขึ้น โดยยังคงไว้ซึ่งประสิทธิภาพที่ดีในระดับเดิมหรือดียิ่งขึ้น

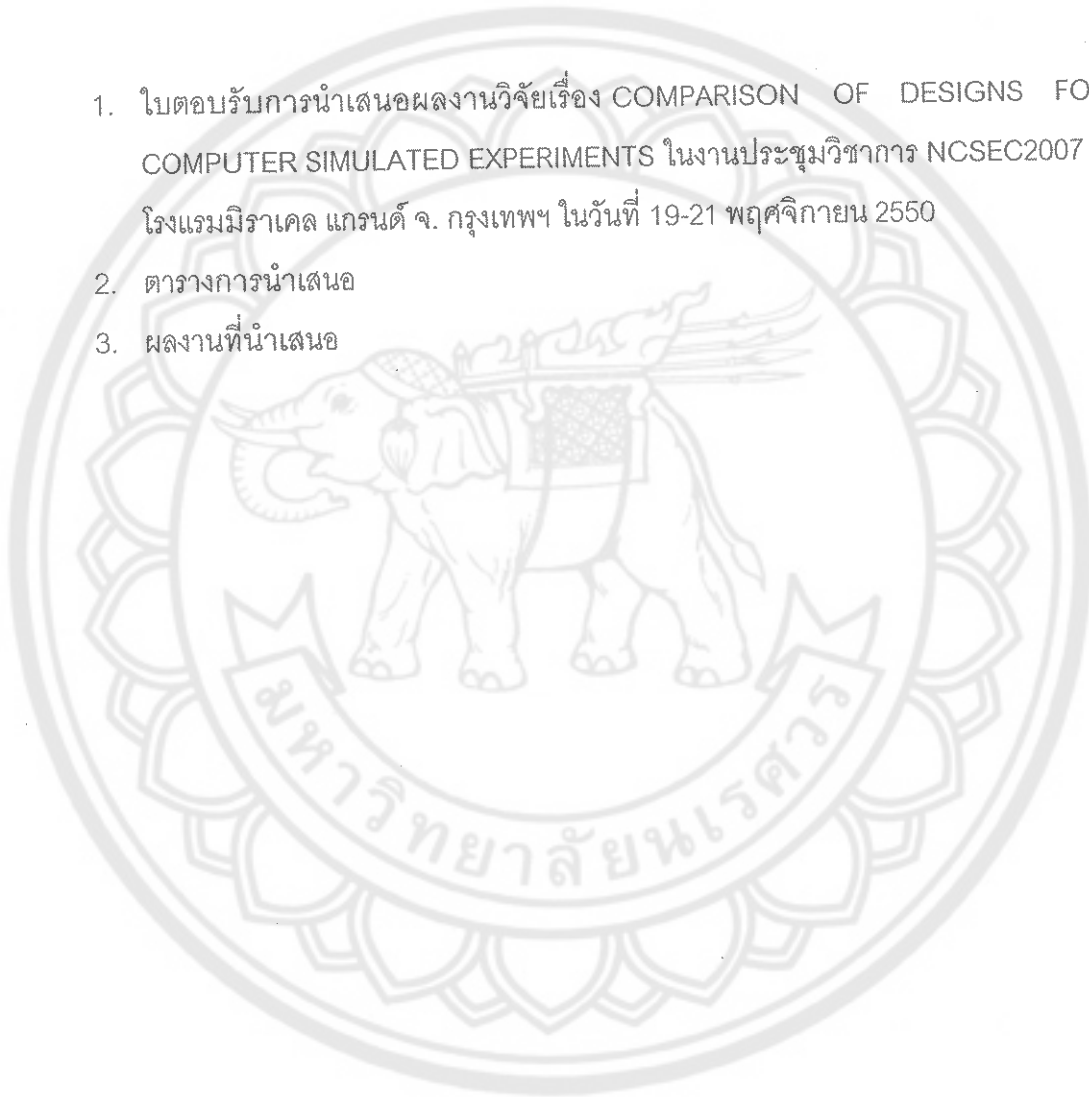
## บรรณานุกรม

- บุญเสริม บุญเจริญผล. (1 พฤศจิกายน 2548). การจำลองเหตุการณ์ (SIMULATION). สืบค้นเมื่อ 27 มิถุนายน 2550, จาก [http://www.krirk.ac.th/education/dr\\_boonserm/simulation.pdf](http://www.krirk.ac.th/education/dr_boonserm/simulation.pdf)
- Eppen, G.D., Gould, F.J., and Schmidt. (1988). *Introduction to Management Science* (2<sup>nd</sup> ed.) Englewood, New Jersey : Prentice – Hall.
- Fang, K., Li, R. and Sudjianto, A. (2005). *Design And Modeling for Computer Experiments*. New York: Chapman & Hall
- Giunta, A. A., Wojtkiewicz, S. F. Jr. and Eldred, M. S. (2003). Overview of modern design of experiments methods for computational simulations. *American Institute of Aeronautics and Astronautics, Aerospace Sciences Meeting and Exhibit* held January 6-9, 2003 in Reno, NV.
- Hillier, F. S. and Lieberman, G. J. (2005). *Introduction to Operation Research* (8<sup>th</sup> ed.). Boston, Massachusetts: McGraw-Hill.
- Jin, R., Chen, W. and Sudjianto, A. (2003). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134, 268-287
- Johnson, M., Moore, L. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26, 131-148.
- Leary, S., Bhaskar, A., Keane, A. (2003). Optimal orthogonal-array-based latin hypercubes. *Journal of Applied Statistics* 30(5), 585-598.
- Liefvendakl, M., Stocki, R. (2006). A study on Algorithms for optimaization of Latin hypercubes. *Journal of Statistical planning and inference* 136, 3231-3247.
- Mckay, M.d., Beckman, R.J., 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, 239-246.

- 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
- Park, J.S. (1994). Optimal Latin-hypercube designs for computer experiments. **Journal of Statistical planning and inference** 39, 95-111.
- R Development Core Team (2005). A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna Austria. ISBN 3-900051-07-0.
- 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.
- Sacks, J., Schiller, S.B. and Welch, W.J. (1989). Design for computer experiments. **Technometrics**, 34, 15-25.
- 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**, Vol.2, No. 3, 209-240.
- Ye, K. Q., Li, W. and Sudjianto A. (2000). Algorithmic construction of optimal symmetric Latin hypercube designs. **Journal of Statistical Planning and Inference**, 90, 145-159.

## ภาคผนวก

1. ใบตอบรับการนำเสนอผลงานวิจัยเรื่อง COMPARISON OF DESIGNS FOR COMPUTER SIMULATED EXPERIMENTS ในงานประชุมวิชาการ NCSEC2007 ที่ โรงแรมมิราเคิล แกรนด์ จ. กรุงเทพฯ ในวันที่ 19-21 พฤศจิกายน 2550
2. ตารางการนำเสนอ
3. ผลงานที่นำเสนอ



# **Proceedings Of The 11th National Computer Science and Engineering Conference (NCSEC 2007)**

*“INVIGORATING THAILAND THROUGH ICT RESEARCH”*

19-21 November 2007  
Miracle Grand Hotel,  
Bangkok, Thailand.

**Organized by:**  
- Sripatum University

**In cooperation with:**  
- IEEE Thailand Section

**Sponsors:**



ORACLE



**jaratsri rungrattanaubol**

**From:** tpcncsec@cp.eng.chula.ac.th [tpcncsec@cp.eng.chula.ac.th]  
**To:** jaratsri rungrattanaubol  
**Cc:**  
**Subject:** NCSEC07 - Notification Letter  
**Attachments:**

**Sent:** พท. 18/10/2550 20:31

Dear Dr. Jaratsri Rungrattanaubol,

On behalf of the NCSEC 2007 Program Committee, we are happy to inform you that the following submission has been accepted to appear at the conference:

Paper Id. and Title: 0127, Comparison of evolutionary search algorithms in computer simulated experiments,  
 Paper Type: Oral

The Program Committee worked very hard to thoroughly review all the submitted papers. Please repay their efforts, by following their suggestions when you revise your paper. The reviews and comments are attached below.

We will soon send you further messages regarding final manuscript preparation, conference registration. Please upload your final manuscript before October 22, 2007, by visiting the following site:

<http://vcr.coe.psu.ac.th/ncsec2007>

Congratulations on your fine work. If you have any additional questions, please feel free to contact us.

--  
 Best Regards,  
 NCSEC 2007  
 Program Committee

===== Additional Information =====

And we would like to inform you that we prepare workshop program about Service ? Oriented Architecture (SOA) Technology on Mon 19th November, 2007 8.30-16.30 at Miracle Grand Hotel. For more information or reservation, please click at <http://ncsec2007.spu.ac.th/>

=====

# >> Session Schedule <<

| 19 <sup>th</sup> November 2007 |       |                                                                                                                                                  |
|--------------------------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 8:00                           | 9:00  | REGISTRATION                                                                                                                                     |
| 9:00                           | 9:15  | กล่าวรายงานโดย : รศ.ดร.สุชาย ธนเสถียร เปิดงานโดย : ดร. มนู อรรถดิลกเชษฐ                                                                          |
| 9:15                           | 10:00 | เรื่อง "สถาปัตยกรรม SOA และขั้นตอนการพัฒนาแบบ SOA"<br>โดย ดร.กิตติมา เมฆาปัญชากิจ สำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (SIPA)             |
| 10:00                          | 11:00 | เรื่อง "การออกแบบชิ้นส่วนให้บริการผ่าน Web (Web Service Object)"<br>โดย อาจารย์ธนิศา เครือไวยวรรณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง |
| 11:00                          | 11:15 | Coffee Break                                                                                                                                     |
| 11:15                          | 12:00 | เรื่อง "SOA Roadmap และการทำ Pilot Project ในองค์กร"<br>โดย รศ.ดร.สุชาย ธนเสถียร คณะบดีคณะสารสนเทศศาสตร์ มหาวิทยาลัยศรีปทุม                      |
| 12:00                          | 13:30 | Lunch                                                                                                                                            |
| 13:30                          | 14:30 | เรื่อง "การแลกเปลี่ยนข้อมูลผ่าน Web Service และมาตรฐาน TH-EGIF"<br>โดย ดร.ชุมพล บุญมี มหาวิทยาลัยธรรมศาสตร์                                      |
| 14:30                          | 15:30 | เรื่อง "Action Research สำหรับ SOA"<br>โดย ผศ.ปกรณ เสงี่ยมสุข มหาวิทยาลัยธรรมศาสตร์                                                              |
| 15:30                          | 15:45 | Coffee Break                                                                                                                                     |
| 15:45                          | 16:45 | เรื่อง "SOA Training "<br>โดย ผศ.ดร.วิวัฒน์ วัฒนาวุฒิจุฬาลงกรณ์มหาวิทยาลัย                                                                       |

| 20 <sup>th</sup> November 2007 |       |                                                                  |                                    |                                      |
|--------------------------------|-------|------------------------------------------------------------------|------------------------------------|--------------------------------------|
| 8:30                           | 9:15  | REGISTRATION                                                     |                                    |                                      |
| 9:15                           | 10:00 | Opening Ceremony                                                 |                                    |                                      |
| 10:00                          | 10:15 | เปิดงานโดย ศ.ดร.วิจิตร ศรีสอาน (รัฐมนตรีว่าการกระทรวงศึกษาธิการ) |                                    |                                      |
| 10:15                          | 10:30 | Coffee Break                                                     |                                    |                                      |
| 10:30                          | 12:00 | Keynote Speaker: James Clark                                     |                                    |                                      |
| 12:00                          | 13:30 | Lunch                                                            |                                    |                                      |
| 13:30                          | 13:50 | รศ.ดร.ครรชิต<br>มาลัยวงศ์ ราชบัณฑิต                              | รศ.ดร.ประกาศ<br>จงสถิตย์วัฒนา      | ศ.ดร.ชิดชนก<br>เหลือสินทรัพย์        |
|                                |       | <i>Data Mining</i>                                               | <i>Decision Support<br/>System</i> | <i>Robotics &amp;<br/>Automation</i> |
| 13:50                          | 14:10 | P123                                                             | P162                               | P128                                 |
|                                |       |                                                                  |                                    | รศ.ดร.ประสงค์<br>ปราณีตพลกรัง        |
|                                |       |                                                                  |                                    | <i>Decision Support<br/>System</i>   |
| 14:10                          | 14:30 | P158                                                             | P138                               | P152                                 |
| 14:30                          | 14:50 | P160                                                             | P163                               | P161                                 |
| 14:50                          | 15:05 | Coffee Break                                                     |                                    |                                      |
|                                |       | <i>Bioinformatics</i>                                            | <i>Cyber Security</i>              | <i>Computer System<br/>Design</i>    |
| 15:05                          | 15:25 | P26                                                              | P17                                | P94                                  |
| 15:25                          | 15:45 | P149                                                             | P91                                | P95                                  |
| 15:45                          | 16:05 | P171                                                             |                                    | P102                                 |
| 16:05                          | 16:25 | P172                                                             | P168                               | P170                                 |
|                                |       |                                                                  |                                    | P31                                  |
|                                |       |                                                                  |                                    | P69                                  |
|                                |       |                                                                  |                                    | P111                                 |
|                                |       |                                                                  |                                    | P130                                 |
| 17:00                          | 18:00 | POSTER SESSION                                                   |                                    |                                      |
| 18:30                          | 22:00 | Reception Dinner                                                 |                                    |                                      |

# >> Session Schedule <<

| 21 <sup>th</sup> November 2007 |       |                             |                                  |                         |                            |
|--------------------------------|-------|-----------------------------|----------------------------------|-------------------------|----------------------------|
| 8:30                           | 9:00  | REGISTRATION                |                                  |                         |                            |
|                                |       | <i>Software Engineering</i> | <i>Computer Networks</i>         | <i>Computer Systems</i> | <i>Information Systems</i> |
| 9:00                           | 9:20  | P0025                       | P0024                            | P0030                   | P0021                      |
| 9:20                           | 9:40  | P0035                       | P0166                            | P0127                   | P0089                      |
| 9:40                           | 10:00 |                             | P0075                            | P0034                   | P0108                      |
| 10:10                          | 10:20 | P0064                       | P0147                            | P0117                   | P0116                      |
| 10:20                          | 10:35 | Coffee Break                |                                  |                         |                            |
|                                |       | <i>Software Engineering</i> | <i>Computer Networks</i>         | <i>Data Mining</i>      | <i>Information Systems</i> |
| 10:35                          | 10:55 | P0079                       | P0049                            | P0022                   | P0153                      |
| 10:55                          | 11:15 | P0085                       | P0074                            | P0028                   | P0159                      |
|                                |       |                             |                                  |                         | <i>Image Processing</i>    |
| 11:15                          | 11:35 | P0090                       |                                  | P0039                   | P0012                      |
| 11:35                          | 11:55 | P0093                       | P0086                            | P0083                   | P0061                      |
| 11:55                          | 13:30 | Lunch                       |                                  |                         |                            |
|                                |       | <i>Software Engineering</i> | <i>Artificial Intelligence</i>   | <i>Data Mining</i>      | <i>Image Processing</i>    |
| 13:30                          | 13:50 | P0118                       | P0154                            | P0099                   | P0068                      |
|                                |       | <i>Neural Network</i>       |                                  |                         |                            |
| 13:50                          | 14:10 | P0047                       | P0112                            | P0142                   | P0082                      |
| 14:10                          | 14:30 | P0051                       | P0125                            | P0144                   | P0106                      |
| 14:30                          | 14:50 | P0055                       | P0139                            | P0165                   | P0140                      |
| 14:50                          | 15:05 | Coffee Break                |                                  |                         |                            |
|                                |       |                             | <i>Artificial Intelligence</i>   | <i>Data Mining</i>      | <i>Image Processing</i>    |
| 15:05                          | 15:25 | P0056                       | P0186                            | P0174                   | P0177                      |
| 15:25                          | 15:45 | P0059                       | P0060                            | P0183                   | P0189                      |
|                                |       | <i>Speech Processing</i>    | <i>Robotics &amp; Automation</i> | <i>Fuzzy Logics</i>     |                            |
| 15:45                          | 16:05 | P0151                       | P0045                            | P0037                   |                            |
| 16:05                          | 16:25 |                             |                                  | P0182                   |                            |
| 16:25                          | 17:00 | Closing Ceremony            |                                  |                         |                            |

# » Contents «

|                                |                                                                                  |     |
|--------------------------------|----------------------------------------------------------------------------------|-----|
| <b>P117</b>                    | A column generation based local search for pickup and delivery problem           | 90  |
|                                | <i>Kriangsak Vanitchakornpong, Krisana Chinmasarn and Nakorn Indra-Payoong</i>   |     |
| <b>P127</b>                    | Comparison of evolutionary search algorithms in computer simulated experiments   | 102 |
|                                | <i>Jaratsri Rungrattanaubol and Anamai Na-udom</i>                               |     |
| <b>P128</b>                    | An Extensive Authentication on GSI for Applications in Grid Services Environment | 107 |
|                                | <i>Suriya U-ruekolan and Jedsada Phengsuwan</i>                                  |     |
| <b>P152</b>                    | The development of a resources sufficiency based scheduling algorithm            | 115 |
|                                | <i>Peerapat Varalertsakul and Putchong Uthayopas</i>                             |     |
| <b>P161</b>                    | Embedding of a Mesh in a Completely Overlapping Network                          | 123 |
|                                | <i>Waricha Yaothanee and Sanpawat Kantabutra</i>                                 |     |
| <b>P170</b>                    | วงจรการจัดการการชนกันของไดนามิกไปป์ไลน์                                          | 133 |
|                                | <i>Paniti Wongpipapong and Arthit Tongtag</i>                                    |     |
| <b>Artificial Intelligence</b> |                                                                                  |     |
| <b>P47</b>                     | การจัดแนวโน้มน้ำหนักสำหรับประมาณด้วยโครงข่ายประสาทเทียม                          | 138 |
|                                | <i>มวิรรณ วาชัยยง</i>                                                            |     |
| <b>P51</b>                     | การพยากรณ์ยอดขายโดยใช้ตัวแบบ Coactive Neuro-Fuzzy Inference System :             | 146 |
|                                | <i>กรณีศึกษา อุตสาหกรรมแหวน</i>                                                  |     |
|                                | <i>จิระนันท์ ดังคะวัชร, สมจิตร อาจอินทร์ และ พัชรี หอวิจิตร</i>                  |     |
| <b>P55</b>                     | ECG and PCG Modeling based on Neural Network                                     | 158 |
|                                | <i>Panee Noiying, Poolsak Koseeyaporn and Chaiyan Suwanchivasiri</i>             |     |
| <b>P56</b>                     | Kernel Fuzzy Radial Basis Function (Kemel-Fuzzy-RBF) Network Classifiers         | 164 |
|                                | <i>Songsook Parakaew, Sirapat Chiewchanwattana and Khamron Sunut</i>             |     |
| <b>P59</b>                     | Selecting a Proper Combination of Learning Algorithms, Activation Functions and  | 172 |
|                                | Error Functions of a Neural Network                                              |     |
|                                | <i>นัฐริษา เหล่าประชา, คำรณ สุนันติ และ สิริภัทร เชื้อขยาญวัฒนา</i>              |     |
| <b>P60</b>                     | Lanna to Thai Language Translation                                               | 180 |
|                                | <i>Nikorn Yaprom</i>                                                             |     |

## Comparison of Evolutionary Search Algorithms in Computer Simulated Experiments

Jaratsri Rungrattanaubol<sup>1</sup>, Anamai Na-udom<sup>2</sup>

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

<sup>2</sup>Department of Mathematics

Faculty of Science, Naresuan University  
Phitsanuloke 65000, Thailand

Email: <sup>1</sup>jaratsrir@nu.ac.th, <sup>2</sup>anamain@nu.ac.th

### Abstract

*Currently the relation of input variables and output response is investigated through the computer simulated experiments (CSE). These experiments are usually time consuming and computationally expensive to run. Therefore a selection of settings of input variables to run CSE is critical. The space filling designs, Latin hypercube designs (LHD) are normally practiced in the context of CSE. The best design for a given problem is obtained by using a search algorithm under a pre-specified optimality criterion. The aim of this study is to compare the capability of two popular evolutionary search algorithms called Simulated annealing algorithm (SA) and Genetic algorithm (GA). The comparisons are made with respect to the optimal value achieved at the termination of each algorithms. The practical guidelines for choice of algorithms to search for the best design (LHD) for some dimensional problems are presented.*

**Key Words:** Computer simulated experiments, Optimal designs, Evolutionary search algorithms, Optimality criteria, Simulated annealing algorithm, Genetic algorithm.

### 1. Introduction

Recently computer simulated experiments (CSE) have replaced classical experiments to investigate a physical complex phenomena, especially when classical (physical) experiments are not feasible. The nature of CSE is deterministic; hence identical settings of input variables always produce an identical set of output response. The process of CSE is typically considered as a black box and not known

a priori. Therefore, space filling designs that aim to spread design points over a region of interest are desired. Space filling designs can be constructed by searching for a design using search algorithms under a pre-specified optimality criterion. This process is based largely on improving the design by exchanging between the pairs of design points.

Latin Hypercube design (LHD) has been extensively used in the context of CSE. It was originally proposed by McKay et al. [4]. LHD is a matrix ( $X$ ), of  $n$  rows and  $d$  columns where  $n$  is the number of runs and  $d$  is the number of input variables. LHD can be constructed based on the idea of stratified sampling [4] to ensure that all subregions in the divided input variable space will be sampled with equally probability. A Latin hypercube sampling has

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

where  $\pi_{ij}$  are the elements of an  $n \times d$  matrix comprising of columns  $\pi_i (i = 1, 2, \dots, d)$ . Each column  $\pi_i (i = 1, 2, \dots, d)$  is independent random permutation of number  $1, 2, \dots, n$  and  $U_{ij}$  are  $n \times d$  values of i.i.d. uniform  $U[0,1]$  random variables independent of the  $\pi_{ij}$ . An example of LHD is presented in Table 1.

**Table 1:** The  $5 \times 4$  random LHD

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

The CSE are usually complex and consist of many input variables to investigate. In this case a large number of runs are required to estimate the parameters corresponding to factors of interest in the model. For example, if the problem of interest consists of  $d$  input variable and  $n$  number of runs are desired, the total number of LHD are  $(n!)^d$ . Obviously this number explodes as values of  $n$  and  $d$  increase, hence full space of LHD cannot be explored. In these cases we need search algorithms to lead us to a good design with respect to an optimality criterion. The key idea of all existing search algorithms is to use some kind of exchange procedures to move towards better designs.

In this paper we investigate and compare the goodness of two popular evolutionary search algorithms, Simulated Annealing (SA) and Genetic Algorithm (GA). Our implementation of these algorithms include some modifications that improve the capability of the search algorithms with respect to the minimization of  $\phi_p$  criteria [5]. The comparisons are made by comparing the optimal values ( $\phi_p$ ) achieved for each search algorithms by using descriptive statistics and Two Independent Samples t-test. Details of SA and GA are presented in the next section, followed by parameter setting, the results and conclusions respectively.

## 2. Research Method

The emphasis of this study is on the comparison of two evolutionary search algorithms namely SA and GA. We first developed each search algorithms in R codes [6] and stimulated them, then the optimal value of  $\phi_p$  criteria for each search algorithms are recorded and the statistical test is performed. The details of SA, GA and  $\phi_p$  optimality criteria are presented as follows.

### 2.1 Simulated annealing algorithm (SA)

Morris and Mitchell [5] adapted a simulated annealing search algorithm to find the best design with respect to  $\phi_p$  criteria. The SA is based on the

idea of exchanging two random points (Perturbation) of a dimension in the design. However in the SA algorithm, an exchange with no improvement or deterioration in the  $\phi_p$  may also be retained during search process.

**Step 1** Set initial value of  $I_{\max}$  (maximum number of perturbation to seek improvement),  $t_0$  (initial cooling temperature) and  $C_t$  (factor by which  $t_0$  is reduced when no improvement in  $\phi_p$  is attained).

The factor  $t_0$  control the probability of retaining a perturbation that results in no improvement of  $\phi_p$ . In particular a higher value of  $t_0$  leads to higher probability of retaining a non-improving perturbation. As search process continues  $t_0$  is changed by a factor  $C_t$  resulting in lower value of  $t_0$  and hence lower probability of retaining non-improving perturbation of same order. Specify the dimension of the problem  $n$  and  $d$ .

**Step 2:** Generate a random LHD  $X$  of given order  $n \times d$ . Let

$$X_{best} = X, t = t_0$$

**Step 3:** Set

$$I = 1, \text{Counter} = 0$$

**Step 4:** Let  $X_{try} = X$

Randomly select a column say  $j$ , of matrix  $X_{try}$  and exchange two randomly selected elements of column  $j$ , say  $X_{aj} \leftrightarrow X_{bj}$ .

**Step 5:** Set  $X = X_{try}$ , Counter = 1,

If  $\phi_p(X) - \phi_p(X_{try}) \geq \varepsilon$  or with probability

$$e^{-[\phi_p(X_{try}) - \phi_p(X)]/t}$$

, when  $\phi_p(X) - \phi_p(X_{try}) < \varepsilon$ .

**Step 6:** If  $\phi_p(X_{try}) < \phi_p(X_{best})$ , set  $I = 1$  and

$$X_{best} = X_{try}, \text{ else } I = I + 1.$$

**Step 7:** If  $I < I_{\max}$  go to Step 4.

**Step 8:** If Counter = 1, set  $t = t \times C_t$ , go to Step 3.

**Step 9:** Stop and report  $X_{best}$ .

### 2.2 Genetic algorithm (GA)

In this study we adopt GA proposed by Liefvendahl and Stocki [3] to search for an optimal LHD. The adopted version has modified method of generating offsprings and generating mutation. GA starts with creating an initial population consisting of  $m$  LHD of order  $n \times d$ . A new generation is created by crossover of LHD's and applying mutations.

The search is continued till stopping criterion is met. The details of the genetic algorithm are presented below.

Notations used for GA are as follows

- $m$  = Population size
- $m_i$  = Number of designs generated in each generation ( $m_i = 10 \times m$ )
- $\omega$  = Mutation rate

**Step 1: Initial population**

Generate  $m$  random LHD of order  $n \times d$ . The initial population comprises of these

$m \{X^i; i=1, 2, \dots, m\}$  LHD.

**Step 2:** Let population  $P$  be  $\{X^i; i=1, 2, \dots, m\}$ , sort LHD with respect to the optimality criterion.

**Step 3: Selecting parents**

Randomly select two LHD from the sequence in Step 1, say  $X^a$  and  $X^b$ . Next randomly select one column from each of the LHD's say  $C_a$  and  $C_b$ .

**Step 4: Generating offspring column**

Once the column  $C_a$  and  $C_b$  are selected. Two new offspring column  $O^a$  and  $O^b$  can be obtained as follows.

- Identify rows in each column having identical elements, copy identical elements in the same row position in column  $O^a$  and  $O^b$ .
- Select two row positions,  $q$  to  $r$ , copy all elements of column  $C_a$  from position  $q$  to  $r$  in offspring column  $O^a$ . Similarly copy the contents of column  $C_b$  from position  $q$  to  $r$  in offspring  $O^b$ .
- Assign remaining entries in column  $O^a$  in sequence of occurrence in column  $C_b$ . Similarly assign the remaining entries in column  $O^b$  in sequence of occurrence in column  $C_a$ .

**Step 5: Generating offspring LHD**

Randomly select a column from each parents LHD  $X^a$  and  $X^b$  and replace it by  $O^a$ . Repeat this process again, this time replace randomly selected

column by  $O^b$ . This process will result in 4 new LHD's.

**Step 6:** Repeat steps 3, 4 and 5 till at least  $m_i$  new offspring LHD's are generated.

**Step 7:** Perform mutations on the collection of  $m_i$  LHD's obtained in Step 6, as follows

- Randomly select a column for mutation with probability  $\omega$ .
- Perform mutation on the selected columns by randomly selecting two points in the column, say  $s$  and  $t$ , and replacing the substring between these points by its cyclic permutation.

**Step 8:** Sort the LHD's generated in Step 7, and retain best  $m-2$  with respect to an optimality criterion, say  $\{X_i^*; i=1, 2, \dots, m-2\}$ . Select two best LHD with respect to optimal criterion in Step 2 and set these as  $X_{m-1}^*, X_m^*$ . The new generation comprises of these  $m$  LHD's ( $\{X_i^*; i=1, 2, \dots, m\}$ ).

**Step 9:** Set  $\{X_i; i=1, 2, \dots, m\} \equiv \{X_i^*; i=1, 2, \dots, m\}$ , repeat Step (2-8) if the stopping criteria as described below is not attained. Otherwise stop the search and report best LHD in the generation with respect to an optimality criteria.

### 2.3 The $\phi_p$ optimality criteria

Morris and Mitchell [5] proposed a modification class of maximin distance criterion in order to search for the optimal design. For a given design  $X$ , let a Euclidean distance list  $(d_1, d_2, \dots, d_m)$ , is the elements list from smallest to the largest. Also define index list  $(J_1, J_2, \dots, J_m)$ , which  $J_j$  is the number of pairs of sites in the design separated by distance  $d_j$ . Thus  $X$  is a maximin design if among available designs, it maximizes  $d_j$  while  $J_j$  is minimized. The scalar criterion can be expressed as

$$\phi_p = \left[ \sum_{j=1}^m J_j d_j^{-p} \right] \quad (2)$$

where  $p$  is a positive integer,  $J_j$  and  $d_j$  specified from  $X$ . The design that minimizes  $\phi_p$  is a maximin LHD in the class. In this study, the adaptive form of  $\phi_p$  [1, 2] which is simpler than equation (2) to implement is considered:

$$\phi_p = \left[ \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{d_{ij}^p} \right]^{\frac{1}{p}} \quad (3)$$

### 3. Data Quality

The capability of each search algorithm is investigated via optimality criteria. The comparison would be made for the dimensionality of the problems as specified in Table 2. For each  $d$ , run size  $n$  are fixed at a maximum number of unknown parameters required in second order polynomial model ( $n = 2d + 4 \binom{d}{2} + 1$ ).

Table 2: Dimension of the problem

| $d$ | 2 | 3  | 4  | 5  | 6  | 7  | 8   | 9   | 10  |
|-----|---|----|----|----|----|----|-----|-----|-----|
| $n$ | 9 | 19 | 33 | 51 | 73 | 99 | 129 | 163 | 201 |

For each dimension of the problem in Table 2, SA and GA are simulated 10 times for specified tolerance and algorithmic parameters with respect to  $\phi_p$  optimality criterion. The differences in average performance of the two algorithms are tested via Two Independent Samples T-test on observed optimality criterion value. This is to find a better search algorithm on average (with respect to  $\phi_p$  optimality criterion).

The two algorithms, SA and GA are implemented via codes written in R [6] to search for an optimal LHD with respect to  $\phi_p$  criteria. The parameter settings for each of these algorithms are presented below.

#### 3.1 Parameter setting for SA

The simulated annealing algorithm requires input parameter settings for  $t_0$ ,  $I_{\max}$ ,  $C_t$  and  $\varepsilon$ . We use the heuristics presented in [5] and our own experience of using SA to set the initial values presented below.

- $t_0$  is set to average Euclidean intersite distance of the initial LHD  $X$ .

$$t_0 = \sum_{i=1}^n \sum_{j=i+1}^n \left[ \sum_{l=1}^d (X_{il} - X_{jl})^2 \right]^{\frac{1}{2}} / \binom{d}{2}$$

- $I_{\max}$  is the maximum number of exchanges before the temperature is reduced. The number should be large enough to allow SA to move away from local minimum. We set

$I_{\max} = 1000$  for  $d \leq 6$ , and  $I_{\max} = 500$  for  $d \geq 7$ .

- $C_t$  is the rate at which the temperature is cooled down.  $C_t$  is set at 0.95, to lower the temperature slowly as the search progresses.
- $\varepsilon$ , the tolerance is set at  $1.0E-15$  for  $d \leq 6$  and as 0.0001 as our empirical studies indicated that setting smaller tolerance levels would not result improvement in optimal value but require longer time to terminate.

In this study we use a modified version of SA to avoid recalculation of  $\phi_p$  values. As SA algorithms require updating of  $\phi_p$  value after an exchange (pairwise exchange) is performed. Every exchange results in change of only two elements of the design matrix and hence two rows (and columns) of the corresponding distance matrix. The resulting new  $\phi_p$  value can be easily obtained by updating old  $\phi_p$  incorporating changes corresponding to changed position of new elements. This process avoids recalculation of  $\phi_p$  at each stage and hence leads to efficient computation. The details of the updating method for  $\phi_p$  can be found in [1].

#### 3.2 Parameter setting for GA

The implementation of GA requires setting of four main parameters namely population size ( $m$ ), number of new designs constructed ( $m_i$ ), mutation rate ( $\omega$ ) and tolerance ( $\varepsilon$ ). Based on the preliminary study of GA, the parameter values were set as follows

$$\begin{aligned} m &= 50 \\ m_i &= 10 \times m = 500 \\ \omega &= 0.05 \\ \varepsilon &= 0.0001 \end{aligned}$$

**Table 3:** Results of  $\phi_p$  optimality criteria where  $d = 2, 3, \dots, 10$ 

| $d$ | $n$ | Algorithm | Min   | Max   | Mean  | SD    | t-test  | P-value |
|-----|-----|-----------|-------|-------|-------|-------|---------|---------|
| 2   | 9   | SA        | 4.274 | 4.350 | 4.324 | 0.033 | -3.716  | 0.002   |
|     |     | GA        | 4.345 | 4.509 | 4.405 | 0.057 |         |         |
| 3   | 19  | SA        | 4.914 | 4.494 | 4.931 | 0.013 | -16.497 | 0.000   |
|     |     | GA        | 5.344 | 5.696 | 5.485 | 0.100 |         |         |
| 4   | 33  | SA        | 5.197 | 5.205 | 5.202 | 0.002 | -37.392 | 0.000   |
|     |     | GA        | 5.670 | 5.820 | 5.760 | 0.045 |         |         |
| 5   | 51  | SA        | 5.415 | 5.423 | 5.419 | 0.003 | -48.790 | 0.000   |
|     |     | GA        | 6.051 | 6.182 | 6.115 | 0.043 |         |         |
| 6   | 73  | SA        | 5.582 | 5.585 | 5.583 | 0.001 | -32.593 | 0.000   |
|     |     | GA        | 6.236 | 6.454 | 6.334 | 0.069 |         |         |
| 7   | 99  | SA        | 5.733 | 5.735 | 5.734 | 0.001 | -52.450 | 0.000   |
|     |     | GA        | 6.246 | 6.341 | 6.289 | 0.032 |         |         |
| 8   | 129 | SA        | 6.166 | 6.197 | 6.185 | 0.011 | -32.191 | 0.000   |
|     |     | GA        | 6.354 | 6.391 | 6.371 | 0.014 |         |         |
| 9   | 163 | SA        | 6.336 | 6.404 | 6.374 | 0.019 | -9.330  | 0.000   |
|     |     | GA        | 6.423 | 6.449 | 6.438 | 0.007 |         |         |
| 10  | 201 | SA        | 6.480 | 6.523 | 6.501 | 0.013 | -11.664 | 0.000   |
|     |     | GA        | 6.565 | 6.642 | 6.609 | 0.024 |         |         |

#### 4. Results

All optimal values of  $\phi_p$  criteria from each dimensional problems stated in Table [2] are presented in Table 3. Descriptive statistics on the  $\phi_p$  values obtained from each search algorithms are displayed in columns 4-7. The columns 4-7 indicated that SA performed much better than GA in terms of minimization of  $\phi_p$  criteria. The SD values shown in column 7 displayed larger amount of variation over 10 replications in GA than SA. This is probably because of the random structure in GA, which is sensitive to different starting points. Whereas SA is based on the neighborhood structure, which moves toward better LHD.

The results in columns 8-9 obtained from Two Independent Samples t-test show that SA is significantly better than GA, as P-values are very small (less than 0.002).

#### 5. Conclusion and discussion

From the results displayed in Table 3, it can be concluded that SA performed much better than GA with respect to the minimization of  $\phi_p$  criteria.

From our empirical studies, we observed that rate of convergence of SA is much faster than GA. Poor performance of GA might be from the method to generate the offspring LHD. Hence other methods could be further investigated to improve the performance of GA. Furthermore, the different dimension can also be further examined.

**Acknowledgements** This research was supported by the Faculty of Science, Naresuan University.

#### 6. References

- [1] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments", *Journal of Statistical Planning and Inference*, 134: 268-287, 2005.
- [2] S. Leary, A. Bhaskar, and A. Keane, "Optimal orthogonal-array-based latin hypercubes", *Journal of Applied Statistics*, 30(5): 585-598, 2003.
- [3] M. Liefvendahl and R. Stocki, "Study on algorithms for optimization of Latin hypercubes", *Journal of Statistical Planning and Inference*, 136: 3231-3247, 2006.
- [4] 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*, 21: 239-246, 1979.
- [5] M. D. Morris and T. J. Mitchell, "Exploratory design for computer experiments", *Journal of Statistical Planning and Inference*, 43: 381-402, 1995.
- [6] R Development Core Team, "R: A language and environment for statistical computing", R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

เลขทะเบียน .....

หนังสือยินยอมการเผยแพร่ผลงานทางวิชาการบนเว็บไซต์

สำนักหอสมุด มหาวิทยาลัยนเรศวร

สำนักหอสมุด มหาวิทยาลัยนเรศวร

เลขที่รับ.....54103638.....

วันที่.....25-07-2554.....

ตามที่ข้าพเจ้า ผศ. ดร. จรัสศรี รุ่งรัตนอุบล (คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์)

ได้ส่งผลงานทางวิชาการ เรื่อง การออกแบบการทดลองคอมพิวเตอร์โดยใช้อัลกอริทึมการสืบค้น

ปีที่พิมพ์ 2550

ข้าพเจ้าขอรับรองว่า ผลงานทางวิชาการเป็นลิขสิทธิ์ของข้าพเจ้า และ ผศ. ดร. เกตุจันทร์ จำปาไชยศรี เป็นเจ้าของลิขสิทธิ์ร่วม เพื่อให้ผลงานทางวิชาการของข้าพเจ้าได้เป็นประโยชน์ต่อการศึกษาและสาธารณชน จึงอนุญาตให้เผยแพร่ผลงาน ดังนี้

- ☐ อนุญาตให้เผยแพร่เฉพาะบทคัดย่อ
- ☒ อนุญาตให้เผยแพร่ฉบับเต็ม (Full text)
- ☐ ไม่อนุญาตให้เผยแพร่ เนื่องจาก .....

กำหนดระยะเวลาในการเผยแพร่

- ☒ 3 ปี
- ☐ ตลอดไป

ลงชื่อ .....

ผู้ช่วยศาสตราจารย์ ดร.จรัสศรี รุ่งรัตนอุบล

วันที่ 21 กรกฎาคม 2554

หมายเหตุ ลิขสิทธิ์ใด ๆ ที่ปรากฏอยู่ในผลงานนี้เป็นความรับผิดชอบของเจ้าของผลงาน ไม่ใช่ของ  
สำนักหอสมุด